U.S. Army Symposium on Artificial Intelligence Research for Exploitation of the Battlefield Environment

November 15-16, 1988
El Paso, Texas

John R. Benton, Editor

Sponsored by

U.S. Army Engineer Topographic Laboratories
U.S. Army Atmospheric Sciences Laboratory
U.S. Army Ballistic Research Laboratory
Project Manager, Smoke/Obscurants

Under the auspices of the

Office of the
Assistant Secretary of the Army for Research, Development and Acquisition

Cohosted by
University of Texas at El Paso
U.S. Army Air Defense Artillery Center and School
TRADOC Analysis Command

Preface

The Army is dramatically changing the ways it makes use of terrain and environmental data. The past is represented by paper maps, acetate overlays and staff weather officers. The Army of the future will be using digital terrain data, geographic information systems, automated meteorological data collection and dissemination, and artificial intelligence software to help tie it all together. A major user of terrain and environmental data will be the Integrated Command and Control (IC^2) systems currently under development. Research efforts on applying Artificial Intelligence to terrain reasoning and atmospheric effects are distributed between the Army, academia and industry. Because of the dispersed efforts, it is difficult, sometimes, for researchers to be aware of some of the relevant research being done elsewhere. These considerations led to bringing together researchers in environmental effects, terrain reasoning, geographic information systems, and the TRADOC schools that have responsibility for establishing doctrine on how the Army will use weapon systems in the field.

Under the auspices of the Assistant Secretary of the Army for Research, Development and Acquisition, the U.S. Army Symposium/Workshop on Artificial Intelligence Research for Exploitation of the Battlefield Environment was held on November 15-16, 1988, in El Paso, Texas. The Workshop/Symposium was sponsored by the U.S. Army Engineer Topographic Laboratories; the U.S. Army Atmospheric Sciences Laboratory; Project Manager, Smoke/Obscurants; and the U.S. Army Ballistic Research Laboratory. The hosting institutions were the University of Texas at El Paso, the U.S. Army Air Defense Artillery Center and School, and the TRADOC Analysis Command. Science Technology Corporation (STC) was responsible for organizing and administrating the symposium and workshop. The efforts of the following STC personnel are gratefully acknowledged: Dr. Ardash Deepak, President of STC, Dr. Paul Try, the symposium moderator and Ms. Carolyn Keene, the conference coordinator, and the members of her staff.

The members of the program committee were as follows:

John R. Benton, Chairman
U.S. Army Engineer Topographic Laboratories

Richard Antony
U.S. Army Signal Warfare Laboratory

MAJ Dave Davis
U.S. Army Engineer School

Robert E. Dekinder, Jr.
U.S. Army PM Smoke/Obscurants

Dr. Philip Emmerman
U.S. Army Harry Diamond Laboratory

Morton A. Hirschberg
U.S. Army Ballistic Research Laboratory

Dr. Howard Holt
U.S. Army Atmospheric Sciences Laboratory

George A. Morales
U.S. Army Air Defense Artillery School

Dr. Joseph H. Pierluissi
University of Texas at El Paso

MAJ Robert E. Richbourg
U.S. Military Academy

MAJ Tom E. Shook
U.S. Army Concepts Analysis Agency

Dr. Kay F. Sterrett
U.S. Army Cold Regions Research and Engineering Laboratory

Dr. Paul D. Try
Science and Technology Corporation

Table of Contents

Session III: State-Of-The-Art Applications

Session IV: Basic Research in Artificial Intelligence

DESIGN OF A SOFTWARE ENVIRONMENT FOR TACTICAL SITUATION DEVELOPMENT

M. J. Coombs and R. T. Hartley
New Mexico State University
Las Cruces, NM 88003-0001, USA

J. R. Thompson
Science Applications International Corporation
Albuquerque, NM 87106, USA

ABSTRACT

This paper concerns the development of a prototype Tactical Situation Development Environment (TSDE) which will aid intelligence analysts to construct and evaluate future situational projections and will support this function by maintaining appropriate models of the current situation. Conventional automated problem solvers have failed in these tasks because they are deterministic, being designed to solve pre-determined problems using pre-defined sets of knowledge and data. They are unable to accommodate the uncertainty characteristic of the realistic battlefield. In contrast, we have employed a general-purpose automated problem solving architecture - Model Generative Reasoning (MGR) - designed for information processing in noisy and ill-specified problem domains. The MGR architecture is currently being developed for a number of military information integration applications, including meteorological data fusion, situation analysis and deception planning.

1. THE SITUATION ANALYSIS PROBLEM

A major part of situation analysis is a data driven process that depends on sensor data for unit and node location and identification. The continuous monitoring of enemy order of battle makes this aspect fairly reliable, being mostly a matter of correlating signatures to equipment and equipment to units through tables of organization and equipment (TO&E). However, analysts tend to become overdependent on sensor data, which commonly leads to operational inflexibility and the susceptibility of intelligence to enemy deception.

Intelligence products with the highest payoff are those that go beyond raw data to predict enemy intentions (Thompson et al., 1988). Models of intentions are valuable because they enable Blue to anticipate Red operations, to identify enemy vulnerabilities, and to improve performance through added preparation time. They are also difficult to construct. The analyst must integrate into a coherent set of

interpretations a complex of diverse, and often dynamic, factors. Moreover, integration has historically been manual, based on sets of map overlays, and must frequently proceed on the strength of very uncertain data, or even the absence of data expected given some observation. Situation analysts thus have a natural tendency to report uninterpreted facts, to hedge bets by qualifying predictions, or to delay predictions until more evidence is available.

Support for intelligence preparation of the battlefield (IPB) is currently supplied in the form of templates. These provide the analyst with prototypical background information on all aspects of the Red and Blue force. Templates represent descriptions of prototypical battlefield elements and are intended to be used as starting points for data interpretation. To them the analyst must add an understanding of the battlefield area in terms of terrain and the effects of weather on friendly and enemy troops, plus any cultural aspects of the area that could impact on operations. Finally, he must integrate battlefield data into this templated picture to form a description of the battlefield situation.

The templating process constitutes a complex hypothesis generation and testing task (Thompson et al., 1983). Situational hypotheses are formed from multiple template overlays which are adjusted to meet environmental constraints. These compound templates are then tested through integration with battlefield data. However, there are strong indications that the templating procedure is not adequate for capturing the complex, highly dynamic and variable events characteristic of a battlefield and so may actually add to the analyst's cognitive load without improving performance (Thompson et al., 1986; Coombs et al., 1988b). In particular, it has been found that: (i) parts of a template my conflict with real intelligence data in ways that require the analyst to engage in a lengthy process of resolution, where the conflict is an artifact of the template's status as a prototype (c.f., Brachman, 1985); (ii) the method does not in itself provide support for resolving incoherence between several different templates selected as interpretations of different subsets of fragmentary data, thus adding to, rather than reducing, the analyst's uncertainty; (iii) an analyst may need to characterize a situation which is novel and which has thus not been templated; (iv) templates do not adequately capture the temporal aspects of situation development.

It is not possible to formulate, test and maintain multiple situational hypotheses of any complexity without automated support. This is particularly true for the analyst working in the field under stress and facing a barrage of changing battlefield data. An automated aid should help the user to: (i) construct and evaluate alternative situation scenarios as competing interpretations of available data; (ii) maintain a set of coherent scenarios in response to changing data; (iii) identify sources of incoherence between the hypotheses and data as they arise; (iv) enable the user to test the effects of tactical assumptions on the set of scenarios in order to develop predictions (i.e., "what if?" games). Moreover, because the automated aid is required to radically enhance analytical skills by changing the user strategies for information integrating, the technology must be capable of being embedded into training systems. Any support system available during training must also be capable of serving as an aid to real situation analysis in the field. The Model Generative Reasoning (MGR) automated problem solving architecture (Coombs and Hartley,

1987, 1988; Fields et al., 1988) developed in the Computing Research Laboratory (CRL) will provide such a technology.

2. TECHNOLOGICAL REQUIREMENTS

Proposals for providing computer support using AI, including automation of the templating method discussed in the previous section, have proved unable to meet the challenge of situation development. In common with other AI-based tactical decision aids (TDAs), automated situation analysis aids employ knowledge-based system technology. This technology was developed by AI for application in complex domains where problems are not amenable to algorithmic solution because of the uncertainty or incompleteness of problem solving knowledge or data. The problem solver must therefore rely on the weaker strategy of edging towards a solution by the application of successive heuristics in response to descriptions of individual problem states. The justification for this approach is clear: a battlefield represents a complex and highly uncertain environment. It is therefore not possible to specify algorithmic solutions, even when problems can be clearly specified in advance. However, the technology has proved to be brittle, i.e., subject to unanticipated failure, in the very environments it was designed to address (Coombs and Alty, 1984).

MGR was designed to provide an architecture for automated problem solving in uncertain task environments. The task environment is defined as *uncertain* when it is not possible to anticipate the range or type of data that the system will have available ahead of time. It will therefore not be possible to identify the set of problems that the system will be able to solve, and hence not possible to specify the knowledge the system will need for solution (Fields and Dietrich, 1988). The objective of an MGR system is to form descriptive structures - *models* - that provide coherent interpretations - *covers* - of particular world situations. Models are generated, evaluated and refined continuously in response to new input from the system's environment. MGR problem solvers are therefore not goal driven in the traditional sense. They have no fixed expectations that determine the structure of solutions, although deviation from expectations may be a factor in the evaluation of solutions for the purpose of answering queries.

MGR provides a powerful technology for determining relationships between uncertain data and templates through its ability to decompose, reconstruct and refine alternative situation descriptions, represented as MGR models, in order to achieve coherent covers of input. MGR is capable of: (i) automatically generating multiple alternative situation descriptions where information is ambiguous; (ii) maintaining the internal coherence of situation descriptions with new input; (iii) enabling a given information item to be viewed in multiple contexts; (iv) decomposing and recombining knowledge structures (e.g. doctrinal templates) in order to cope with novel conditions; (v) projecting situations in time. Furthermore, the MGR architecture is intrinsically parallel. Systems developed using MGR are thus ideally suited for implementation on fast parallel hardware.

3. THE MGR ARCHITECTURE

The MGR architecture is shown in Fig. 1, and is described in Fields et al. (1988). MGR is logically a shared-data parallel virtual machine that accepts input from two databases, a fact database **F** containing input data and a definition database **D** containing stored knowledge. Models are generated by combining information from **D** with information from **F**. Four operators, *specialize* (**Sp**), *fragment* (**Fr**), *merge* (**Mr**), and *generalize* (**Gn**), act on the population **M** of models in an autonomous fashion. The functionality of these operators is specified completely by the architecture. The activity of the operators is governed at a control level above them; control determines when the operators act, but not their functionality. Strategy in MGR thus consists largely of scheduling these four operators. Additional operators *select* (**Sl**) and *evaluate* (**Ev**) are employed to regulate the flow of facts from **F** and definitions from **D** to **M**, and to evaluate the resulting models with respect to user-specified criteria, respectively. The precise functionality of these operators, unlike that of the graph manipulation operators, is specified as part of each application program.

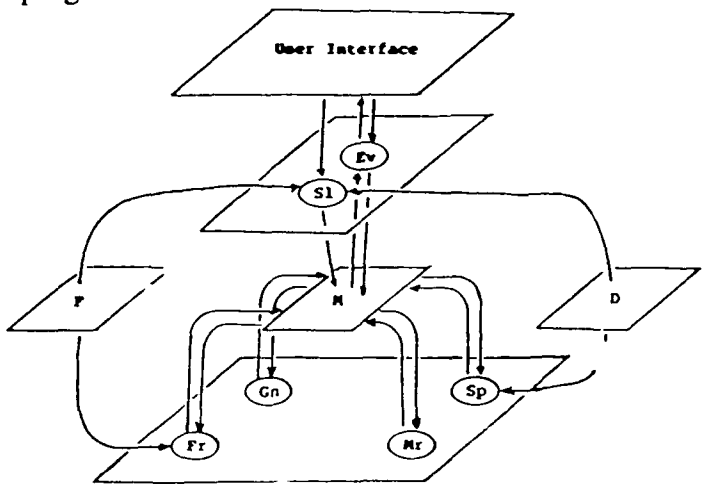


Figure 1. Data-flow diagram of the MGR architecture.

Informally, **Sp** takes a model **m** as input, and generates a set of larger, more specialized models by adding definitions. The role of **Sp** is, therefore, to "glue" knowledge to facts to create models that cover the facts. The number of alternative models generated by each cycle of specialization may be changed by regulating the selection of the subset **D′** of definitions used as input to **Sp**. In the current implementation of **Sp**, **D′** is chosen as a minimal subset of **D** that allows a complete cover of **m**; **Sp** thus generates parsimonious models. **Fr** opposes **Sp** by breaking models into fragments in a way that preserves the information contained in facts, but may destroy some relational information obtained from definitions. **Fr** thus removes definitional information, but not factual information; its role is to break apart models that do not cohere with the available facts in order to generate fragments that can be recombined. Like **Sp**, **Fr** can be regulated through choice of the input fact set **F′**; **F′** is taken to be **F** in the current implementation. **Gn** and **Mr** take subsets of models as input, and generate single models as output which are, respectively, less or more specialized than the models from which they were

generated. **Gn** is capable of generalizing both factual and definitional information; its role is to maintain coherence by removing over-specializations. **Mr** merges models whenever possible; its role is to generate models that have the greatest possible covering power. All of the operators write over the model population; the set of models available for operating on, therefore, changes after every operator application.

4. THE TACTICAL SITUATION DEVELOPMENT ENVIRONMENT (TSDE)

4.1 THE TSDE CONCEPT

The concept of a Tactical Situation Development Environment (TSDE) for analysts engaged in situation development arose from research undertaken into intelligence information integration by (Thompson et al., 1983), in co-operation with USAICS, and by (Thompson et al., 1986), in co-operation with the RADC. These studies concluded that the greatest single limitation is that human analysts are only able to consider one hypothesis at a time. The TSDE provides a basic set of data interpretation and situation modeling capabilities to be used as an operational aid to help analysts maintain and evaluate multiple hypotheses.

The TSDE architecture supports two contrasting analytical functions: (i) the maintenance of baseline situation descriptions in response to new intelligence information, and (ii) the projection of selected situation descriptions in order to evaluate hypotheses concerning possible Red intentions. In baseline processing the objective is to generate situation descriptions that are maximally faithful to available intelligence data. In projection processing the objective is to generate descriprions that are maximally faithful to user assumptions.

The baseline situation development module interfaces to a database of intelligence reports. Data related to a specific mission task are selected from the report database and submitted to MGR for processing as sets of MGR facts **F**. The data are then interpreted using environmental information and deployment/activity templates, represented as the MGR definitions **D**, to form a set of alternative descriptions of the current situation, represented as the set of MGR models **M**. The alternative descriptions are then evaluated as explanations for the data in **F**. Further development of models is driven either by weaknesses discovered during evaluation, the arrival of new reports, or by user queries.

The projection module enables a user to construct temporal extensions of baseline models in order ask hypothetical questions con erning the development of events and their military impact in terms of threat, risk and uncertainty. Projection will usually focus on specific hypothetical ("what if?") queries which require the introduction of query specific assumptions into models, e.g., the examination of possible avenues of approach may require the addition of assumptions concerning Red's previous use of available avenues. It is unlikely that such information on Red's past actions would have been brought in by data during the construction of baseline descriptions and so it will have to be added in response to the query before predictions of possible Red actions can be generated. Other queries may require the selection of subsets of baseline models, e.g., those coherent with some query assumption, or the extraction of specific features, e.g., the removal of all

environmental factors in order to give solutions a doctrinal perspective.

4.2 REPRESENTATION

Intelligence reports, deployment/activity templates and situation descriptions will be represented through the Conceptual Programming (CP) system (Hartley, 1986) as sets of conceptual graphs (Sowa, 1984). CP is being employed for knowledge representation in the MERCURY meteorological data fusion system (Coombs et al., 1988a) and has the capability of mixing the three levels of constraints between objects: structural relations, temporal constraints and constraints qualitative/quantitative relations.

In contrast to the static prototypical templates currently used for situation development, TSDE will form dynamic structures that integrate the material and functional aspects of military units with environmental factors. These structures are also capable of being animated and so initial states may be projected through time and space to generate a sequence of snapshots representing an evolving event. Any uncertainties in the development of events will become visible as underdetermined inputs to procedural and functional constraints. The example presented later demonstrates these capabilities.

4.3 TSDE PROCESSING MODULES

4.3.1 Baseline Module

The objective of the baseline module is to maintain a set of models that give the most complete and detailed description of the current situation that is coherent with available data. Interpretations are generated to the extent of naming Avenues of Approach (AAs) and indicating Named Areas of Interest (NAIs). The templates transferred to the set of working definitions **D** thus serve as a set of expectations which are used to augment and amplify relations between data items.

A basic control strategy has been adopted for the baseline module that ensures that baseline models are faithful to current data. This is achieved by generating specialized models which are complex and so are defeasible by new information. This will maximize the likelihood of a model being negated by new input. Following the basic diagnostic strategy reported in (Coombs and Hartley, 1987), this is implemented by giving the operators *specialize* (**Sp**) and *merge* (**Mr**) preference over *fragment* (**Fr**) and *generalize* (**Gn**). **Sp** will generate larger, more specialized models by adding definitions, while **Mr** will integrate them into maximally related structures. **Gn** is applied when an incoherence is found between unsupported concepts during the application of **Mr**. This has the effect of allowing the **Mr** to execute by generalizing away the blocking concepts. Since these concepts have no direct support from data, there is no reason to maintain their level of precision.

4.3.2 Projection Module

Projection is undertaken on available baseline models in order to address specific intelligence queries. Queries typically focus on one of six questions concerning possible future enemy dispositions of actions. These include: who?, what?,

where?, when?, with what?, and with what strength? Responses require the generation of hypothesized situation projections that elaborate alternative AAs and NAIs to give Target Areas of Interest (TAIs). The set of projections can then be evaluated with reference to the degree of enemy threat, the risk to friendly forces and levels of uncertainty. The projection module will seeks to project baseline models around opposing, although not necessarily incoherent, query assumptions. The projection strategy will seek to maximize the effects of these assumptions and the query will be answered over the resulting set of projected models.

The basic model generation strategy for the projection module varies from that of the baseline module in its emphasis on elaborating differences between models that impact on a query, irrespective of whether they have factual support. Following an initial run of *specialize* (**Sp**) to fold in user assumptions, *merge* (**Mr**) and *generalize* (**Gr**) will only be applied under user direction to support "what if?" experiments; **Gn** will be used to test the effects of loosening constraints imposed by specialized, but unsupported, sections of model while **Gn** will be used to test the coherence of model fragments. Currently Evaluation is under user control.

5. THE BATTLE OF EDGEHILL

5.1 AN HISTORICAL ACCOUNT

The battle of Edgehill was the first real encounter in the English Civil War between King Charles 1st's army and the opposing Parliamentarians. On 12th. October 1642, the King was in Shrewsbury ready to march on London. The Earl of Essex, who led the Parliament's army, was stationed at Worcester. The King started to march to London hoping that he could reach London before Essex, or at least encounter the enemy and beat them in the field. Another factor in the decision was that the King favored a cavalry battle, and the land around Worcester was 'enclosed' i.e. the land was parcelled up and divided by thick hedgerows - unsuitable for cavalry. October was almost too late for a serious campaign since the nights get too cold, and the roads become impassable.

After ten days, the King reached Banbury, threading between Parliament garrisons at Coventry and Warwick. Meanwhile Essex started from Worcester on the 19th. having got news of the King's march, and reached Kineton on the 22nd. The Royalists decided to rest at Edgecote and planned to send out a small brigade to take Banbury. The King scattered his army around the surrounding villages for the night, and it was in one such village that the King's quartermasters encountered those of the enemy. Having received this unexpected intelligence, the King sent out a force to confirm their presence in Kineton. The King resolved to rendezvous at Edgehill the following day, while Essex, acting on intelligence about the King's intent to march on Banbury, decided to try to relieve the town. The battle was then inevitable, since Edgehill lies on the path from Kineton to Banbury.

Edgehill was a steep escarpment devoid of trees, and the King assembled his troops on the top. Essex did not seem inclined to attack up the hill, so the King decided to go down and attack him. The King's cavalry, led by his nephew Prince Rupert, routed their opponents on the flanks, chasing them back to Kineton, but the foot soldiers fought to a draw, both sides being exhausted by the end of the day.

Essex took his army off to Warwick, leaving the King a tactical victor. See Fig. 2 for a map showing the critical towns and the AA's.

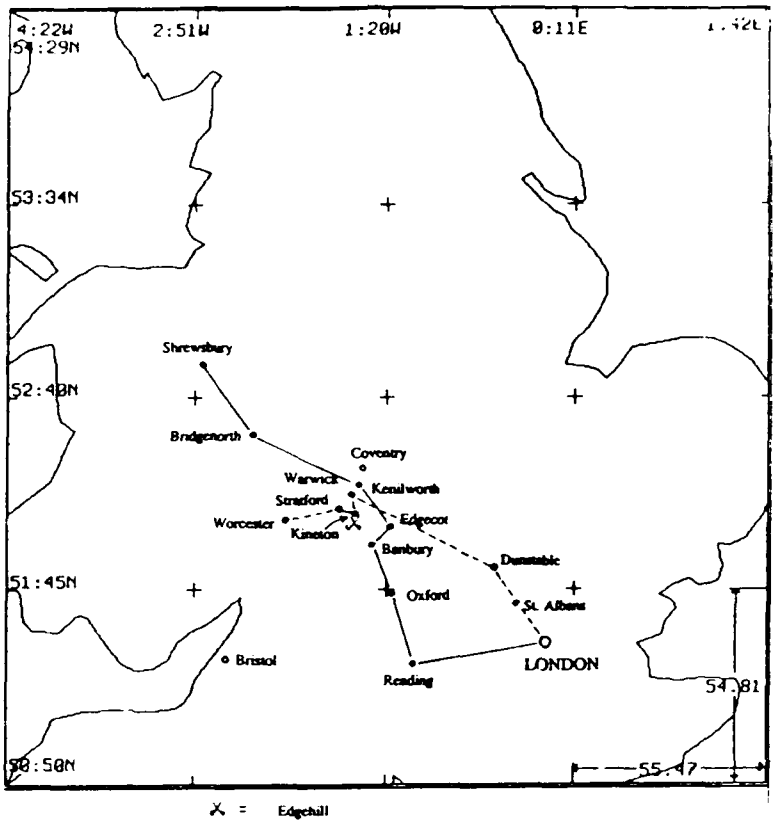


Figure 2. The theatre of war.

5.2 OUR HYPOTHETICAL ACCOUNT

We have taken the Parliamentarian's point of view and have modified the historical account somewhat in order to display how the decision making might have gone with better intelligence gathering. Our intent is to show capabilities in providing automated help for the four objectives outlined in Section 1. They are reproduced here for easy reference:

(i) to construct and evaluate alternative scenarios as competing interpretations of available data;

(ii) maintain coherence in the scenarios in response to changing data;

(iii) identify sources of incoherence between hypotheses and data;

(iv) enable the user to test the effects of tactical assumptions to develop predictions.

We will show that our example of the battle of Edgehill demonstrates capability in each of these areas.

5.2.1 Phase one: Baseline Models

The generation of the initial set of baseline models fulfills objective 1 above. We also show that by changing the input data only slightly, the nature of the models generated can change radically. This covers the second objective. We will now show, in detail, how these different scenarios are produced by MGR. Throughout the example, we take the Parliamentarian's point of view. Thus the 'Red' army are the Royalists, led by King Charles.

5.2.1.1 Available Intelligence

Initially there are three pieces of intelligence. They are:

INT1: the date - 9/20/1642;
INT2: the location of the King's army - at Shrewsbury;
INT3: a leading indicator - the fact that the King's army has been seen to be "victualling" (gathering food and supplies) at a low level of activity. As we shall see, this is evidence that the King intends to garrison Shrewsbury, possibly because of the lateness of the date in the campaign season, and the coming of winter.

We assume that intelligence has been gathered, filtered for immediate relevance (although MGR is capable of accepting any input) and converted to conceptual graph (CG) form. They are then input to the CP knowledge engineering environment, in the form of CP *facts*. These facts are shown in CG form in Fig. 3. They are clearly unconnected, in the sense that they have no labels in common. MGR's job, initially, is to connect them by finding a suitable minimal combination of definition graphs that covers all the labels in the fact set. To look at the possibilities of cover, we need to look at the available definitions.

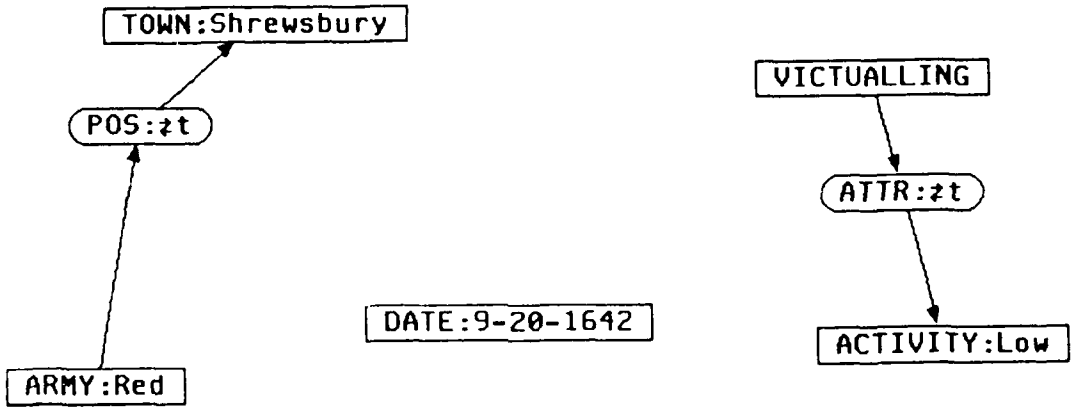


Figure 3. The initial pieces of intelligence

5.2.1.2 Knowledge of Red's Possible Tactics

All definitions are also input into CP and are then available to MGR. MGR makes connections between facts and definitions by matching the labels in the fact set with the contents of all definitions. Where a match is found, a partial cover of that fact with the corresponding definition exists. Total coverage is obtained when all labels in the fact set are covered by at least one definition. Since alternative combinations of definitions may be found to provide complete coverage, then a minimization is carried out, resulting in the minimum sets of definitions that cover the facts. If a particular label in a fact has only a single definition that covers it, then that definition is guaranteed to be necessary for complete cover. When a label has many alternative definitions that cover it, then, in general, this will produce many alternative total covers. The first is what happens initially in our example. The label VICTUALLING is only found in the definitions of the tactical options called GARRISON and PREPARE.

5.2.1.3 The GARRISON option

The graph for the definition of GARRISON is shown in Fig. 4. It contains the following information. An army garrisons in a town at the start of winter, and its leading indicator is the level of victualling activity. A *constraint actor*, called WINTER?, will fail a model containing it unless it is the start of winter, and the level of victualling activity is "low". This is an example of the three layers of knowledge represented in CP. The first layer simply relates terms together; the second, the procedural overlay, relates any states and events together in a temporal fashion; the constraint overlay conditionally relates any instantiations of the term labels (numbers, dates, names etc.). This is a highly simplified account of what it was for an army to garrison, but it illustrates the principles of the representation of such knowledge. If there are multiple accounts of garrisoning that are possible, CP can also represent these for MGR to incorporate into its models.

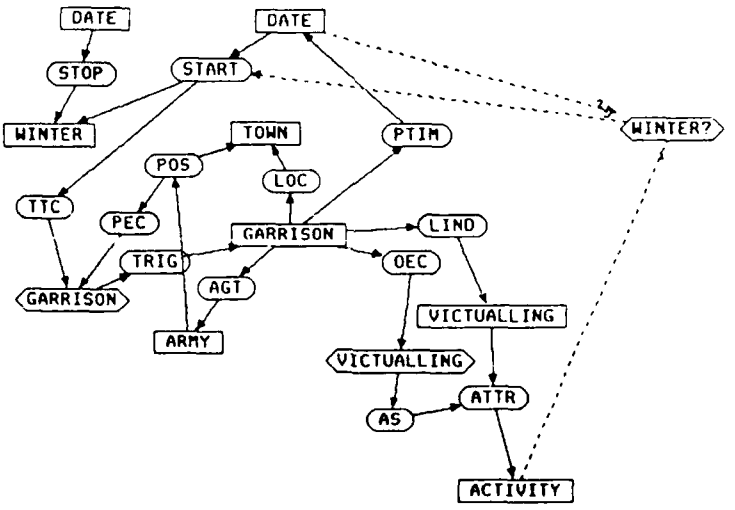


Figure 4. The definition of GARRISON

5.2.1.4 The PREPARE option

The alternative to garrisoning for an army is to prepare to march. Its definition is very similar to garrison, but does not mention winter. A model containing it will only succeed if the level of activity is "high". (The distinction between high and low could be made more elaborate, but here only a binary distinction is necessary).

5.2.1.5 The first models.

With the facts INT1-3, three models are produced, two with GARRISON, and one with PREPARE. Only the GARRISON model succeeds, however, because the constraint actor in PREPARE fails, and in the other GARRISON model, the fact's date is identified with the end of winter instead of the start. The successful model is shown in Fig. 5.

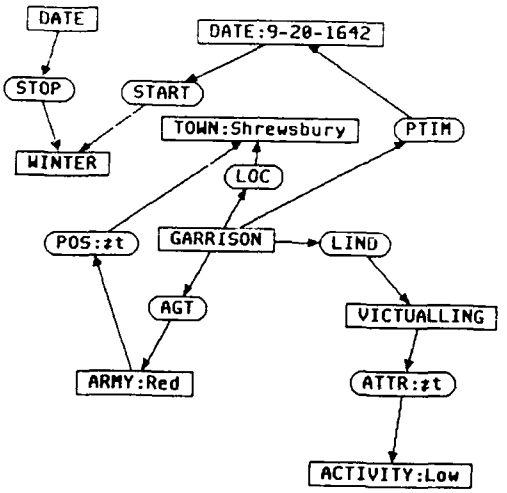


Figure 5. The GARRISON model.

At this point we assume that new intelligence reaches us, raising the level of victualling activity to high. This is INT4. The date has also progressed to October 13th. (INT5). By swapping INT4 for INT3, and INT5 for INT1, the the situation becomes inverted. The GARRISON models fail, leaving the PREPARE model.

There is in both cases a single model that covers all the selected facts. Now assume that intelligence is gained of Red's new location (at Bridgenorth), and of the King's *mission*. This should give us a way of assembling models that explore the various possibilities of Red's objectives, and their consequences.

5.2.1.6 Red's Mission

From Fig. 6 it is clear that the King plans to take London before November 11th., and that Rupert, the King's cavalry commander is a major player. Any models that take this into account will need to talk about battle tactics rather than the concerns of feeding the army that were incorporated into the first models. We have assumed three possibilities here. Firstly, there is the possibility of a BLITZ march on London, as fast as possible, presumably travelling without heavy artillery

which would slow the march considerably. Secondly, Red could march to a friendly large town (Bristol was the obvious choice) where it could REINFORCE its ranks, then march on London. Lastly, Red could engage the enemy in the FIELD, beat them, and then march on London at its leisure. These three options are represented with temporal and constraint information in as generic a fashion as possible. Indeed they may well be Blue's possible tactics as well, but we should assume that they are Blue's understanding of Red's preferences, and do not necessarily have any wider significance.

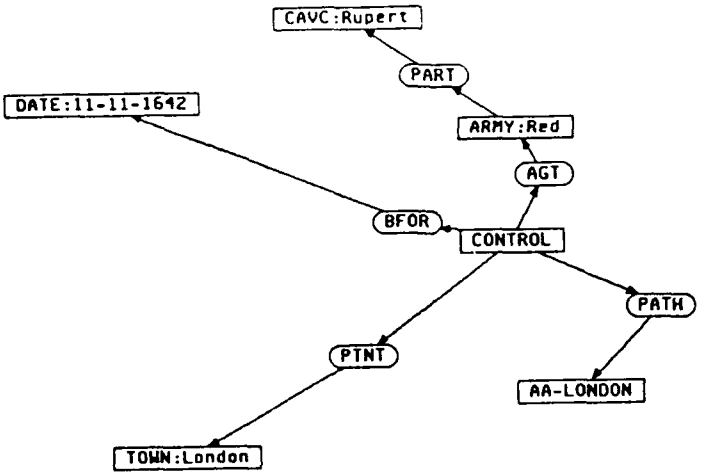


Figure 6. *The King's mission.*

With the new information, nine models are produced, three for each of the battle tactics combined with three possible avenues of approach (through Bristol, through Oxford, and through St. Albans). Of these nine, only two survive, the others being rejected on constraint satisfaction grounds. For instance, all three avenues of approach through Bristol are rejected because the march would take too long. The two surviving possibilities are to BLITZ through Oxford, and to engage in a FIELD battle. The latter survives because Rupert is a cavalry commander, and would prefer a FIELD battle in open terrain. The FIELD model is shown in Fig. 7.

Because there are two models which cover the latest intelligence, MGR can attempt to merge them in order to combine the previously separate options. In this case, the merge succeeds, resulting in a BLITZ/FIELD model which show how a battle could occur during a fast march on London through Oxford.

5.2.2 Phase two: Projection of Blue's Assumptions

In the second phase of situation analysis, the baseline models are assumed to have been evaluated and accepted as probably representative of Red's intentions. The next phase - projection - tests the coherence of a Blue assumption with any of the baseline models. In the example we have encoded two such assumptions (Fig. 8). The intent is to show how the named areas of interest in the baseline models (the towns on the various avenues of approach) become target areas of interest in the projection models. In particular, the second assumption leads to the projection

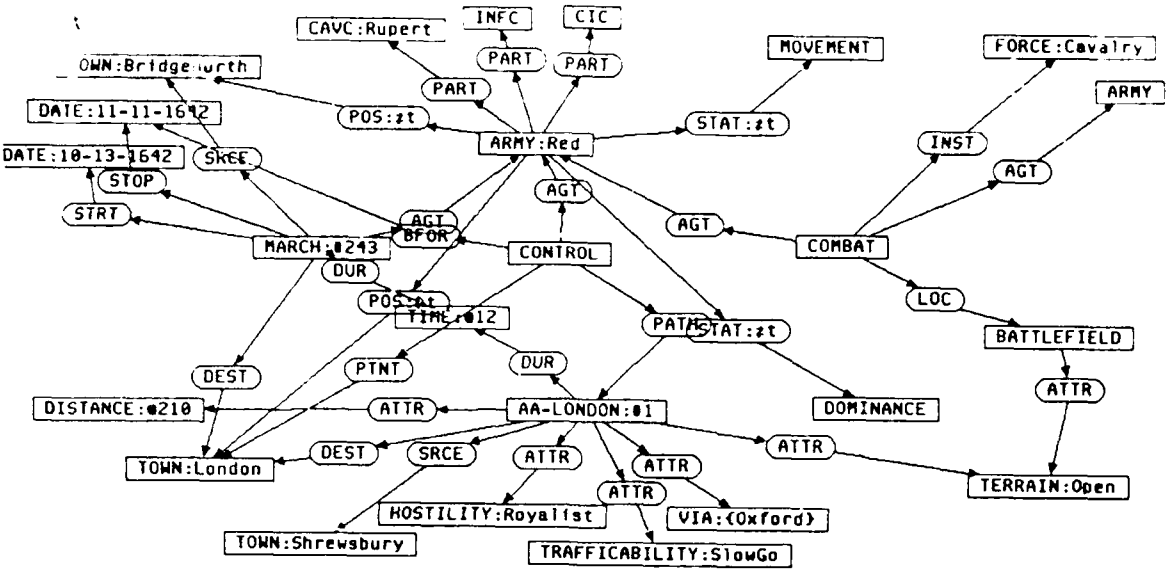


Figure 7. The FIELD model.

of a battle site (Edgehill itself) as a target area.

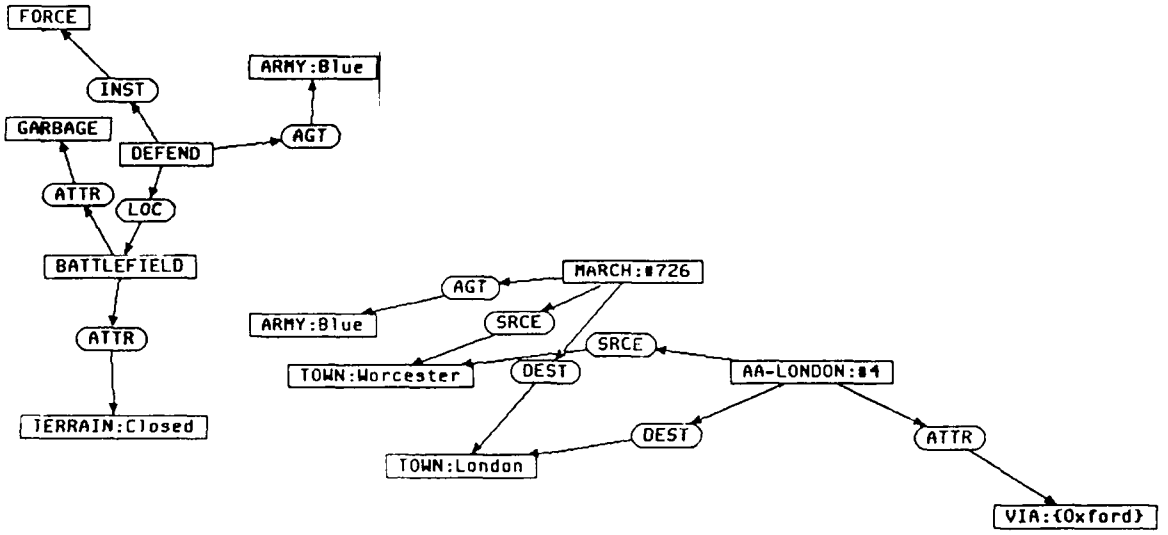


Figure 8. Blue's two assumptions

The first assumption, however, produces no such additional information. The assumption that Blue will march from Worcester to London via Oxford does not clash with the models of Red's likely choices. The interpretation of this result is that the assumption made is too weak.

5.2.2.1 The Choice of a Battle Site

Thge second assumption is purposely stronger. It states the desired intention of Blue to defend any battlefield chosen by Red with a particular type of force, and that the battlefield should be in Closed terrain (i.e. unsuitable for cavalry), and be flat land. This assumption is coherent with the BLITZ model, producing little of

interest, but is incoherent with the FIELD model. This incoherence triggers MGR into generalizing the assumption, since it is assumed to be too strong. The generalization, when re-specialized with the intelligence facts that the model was built on, produces a model with the definition of a *any* battle joined in which, as its constraint overlay, chooses a battle site by correlating the set of towns on the avenue of approach, and the type of terrain. In this case, the terrain is forced back to 'Open', and the appropriate battleground is found to be Edgehill (Fig. 9).

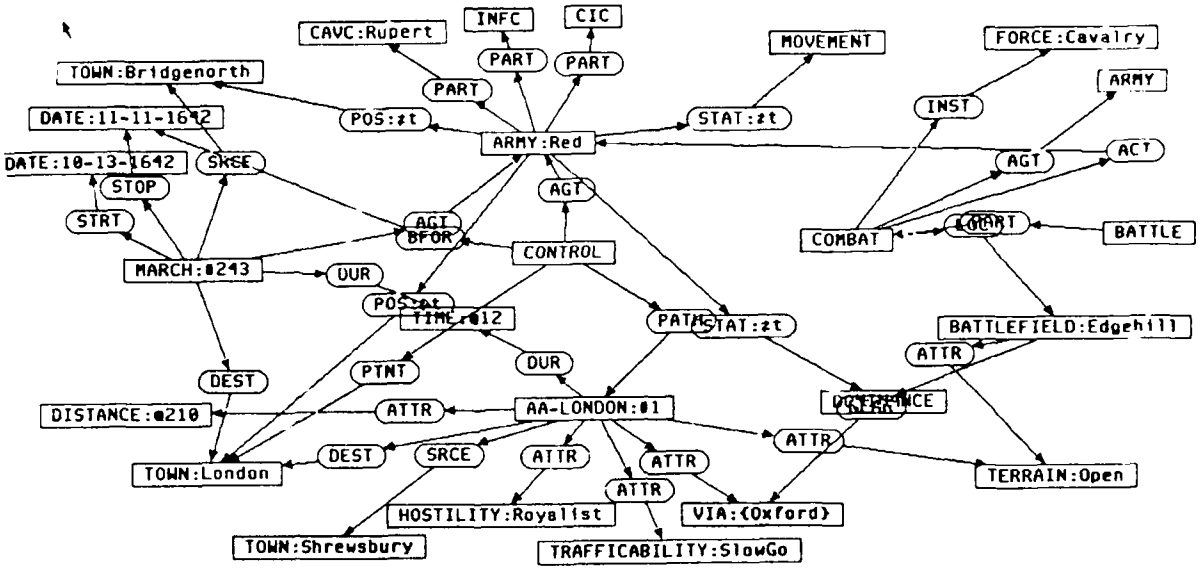


Figure 9. The final model

REFERENCES

Brachman, R. J., 1985: "I lied about the trees", or, Defaults and Definitions in Knowledge Representation, AI Magazine, Fall 1985, 80-93.

Coombs, M. J., and J. L. Alty, 1984: Expert Systems: An Alternative Paradigm, International Journal of Man-Machine Studies, 20, 21-44.

Coombs, M. J., and R. T. Hartley, 1987: The MGR Algorithm and its Application to the Generation of Explanations for Novel Events, International Journal of Man-Machine Studies, 27, 679-708.

Coombs, M. J., and R. T. Hartley, 1988: Explaining Novel Events in Process Control through Model Generative Reasoning, International Journal of Expert Systems, 1, 89-109.

Coombs, M. J., C. A. Fields, and G. McWilliams, 1988a: Artificial Intelligence Methods for Optimizing the Use of Meteorological Databases: Recommendations for Implementing the MERCURY System, ASL Report CR-88-0034-2, US Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM 88002-5501.

Coombs, M. J., J. R. Thompson, R. T. Hartley and T. Fichtl, 1988b: A Software Environment for Tactical Situation Analysis, White Paper, Computing Research Laboratory, New Mexico State University, Las Cruces, NM 88003-0001.

Fields, C. and E. Dietrich, 1988: Engineering Artificial Intelligence Applications in Unstructured Task Environments: Some Methodological Issues. In D. Partridge (Ed.), Artificial Intelligence and Software Engineering, Erlbaum, Hillsdale, NJ.

Fields, C., M. J. Coombs, and R. T. Hartley, 1988: MGR: An Architecture for Problem Solving in Unstructured Task Environments, Proceedings of the Third International Symposium on Methodologies for Intelligent Systems, Elsevier, Amsterdam, 44-49.

Pitz, G. F., and N. J. Sachs, 1984: Judgement and Decision: Theory and Application, Annual Review of Psychology, 35, 139-163.

Sowa, J. F., 1984, Conceptual Structures, Addison Wesley, Reading, MA.

Thompson, J. R., R. Hopf-Weichel, and R. E. Geiselman, 1983: The Cognitive Basis for Intelligence Analysis, OSD Report No. R83-039C, LOGICON Inc., Woodland Hills, CA 91367.

Thompson, J. R., R. Trout, and B. Landee-Thompson, 1986: Artificial Intelligence Applications for Sensor Data Fusion, Report A002, Perceptronics, Woodland Hills, CA 91367.

Thompson, J. R., B. Landee-Thompson, T. C. Fichtl, and L. Adelman, 1988: Measurement and Evaluation of Military Intelligence (MI) Unit Information Processing Performance: Year 1 Technical Report, Report No. 88/6504, Science Applications International Corporation, Alexandria, VA 22333-5600.

ISSUES SURROUNDING DEVELOPMENT OF METEOROLOGICAL EXPERT SYSTEMS

For DoD Use

Timothy Sletten and Mark Stunder
GEOMET Technologies, Inc.
Germantown, Maryland 20874, U.S.A.

ABSTRACT

The purpose of this paper is to describe several meteorological ES's developed for DoD use by examining the issues surrounding their development. The first issue is the need for the ES's. Reasons are presented which show how these systems can benefit DoD decision makers. These reasons include consolidation of knowledge and technology transfer. A second issue is knowledge acquisition. Proper knowledge is the key to the development of any ES and is especially important in the creation of meteorological systems. We discuss how knowledge should be acquired in relation to available experts. We also argue that knowledge acquisition goes beyond just a discussion with an expert, but instead includes detailed examination of literature and data. Pitfalls within the knowledge acquisition process will be discussed. The final issue is user acceptance. User acceptance is examined from two perspectives: quantitative (statistical) data designed to show how the ES measures up to an observed data set and more importantly, qualitative, where the ES is rated by the user. Specific examples of user acceptance are discussed. This includes showing actual evaluation sheets designed to aid in the qualitative evaluation process.

1.0 INTRODUCTION

The expert system area continues to show increasing popularity and promise for use in environmental decision making and in meteorology. A Knowledge Based Expert System (KBES) (see Fig. 1) is basically a structured collection of knowledge that can interact with users. This interaction is accomplished by a series of questions and answers or by a series of data inputs directly into the system. These questions or data inputs are in a controlled sequence that is designed to access the knowledge data base. The end product results in the user receiving recommendations with a probability of success. Expert systems also include the capability to describe their line of reasoning and to play "what if" games with various data input.

We will begin with a brief overview of some primary advantages and justification for the expert system. Next we will look at various steps and some important associated issues in developing

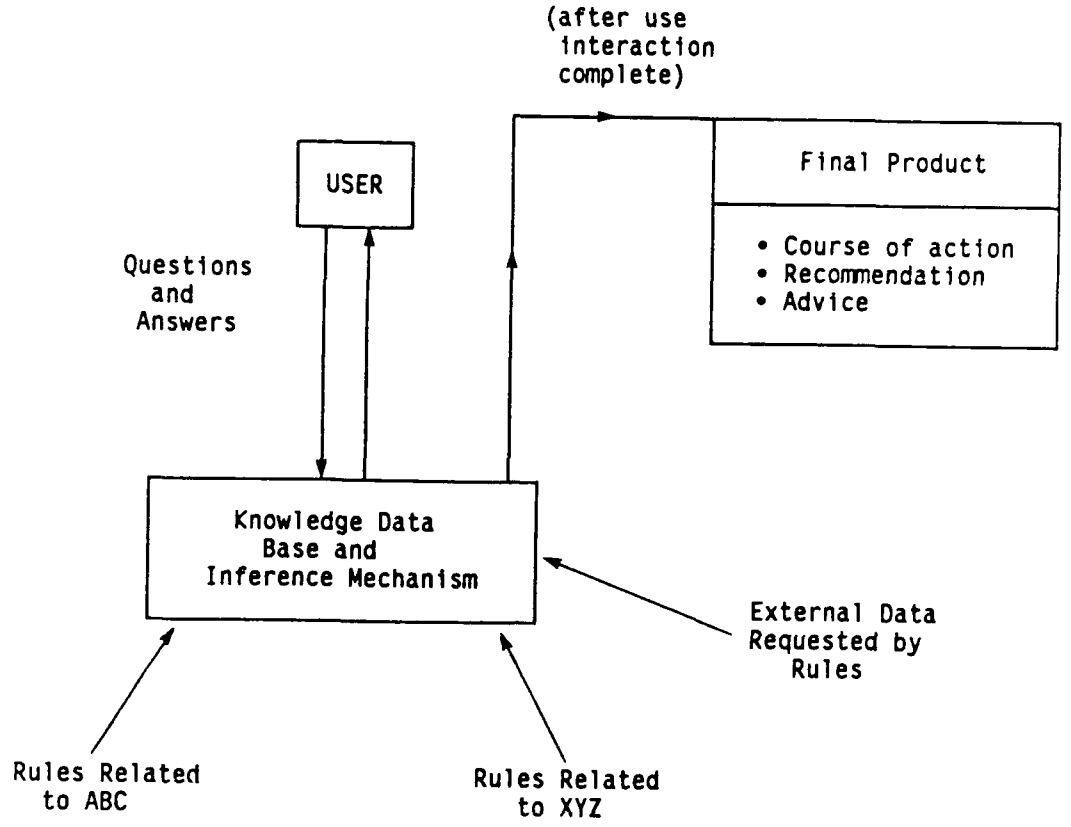


Figure 1. Generalized structure of an expert system.

these systems. In doing this we will relate some of these issues to the systems we developed for DoD.

2.0 ADVANTAGES AND JUSTIFICATION FOR THE EXPERT SYSTEM

Much of the justification and reasoning for an Expert Advisory System can be simply traced to its differences from more traditional numerical-model-related, output oriented programs. These differences can be broken into three key areas:

- Representation of information (data/knowledge)
- Processing
- Explanation.

Numerical models manipulate data only; they are totally dependent upon the right initialization of this data. A KBES deals with heuristics and knowledge (along with the data needed to use the knowledge). In an expert system we are able to represent the total meteorological picture using rules of thumb or structures that are not directly encumbered by number crunching.

Expert systems have explanatory facilities that can explain both their lines of reasoning and individual rules. An expert system knowledge data base also has the capability to be readily changed, should new information become available.

In addition, these systems act as valuable training tools. Using simple English explanatory terms, these systems can teach and illustrate for the user the local factors to be considered when evaluating a problem or making a forecast. This is important in decreasing the orientation time for the new man on the job or the new user in the field.

There is also great value to capturing and preserving knowledge. As technology/techniques, people and priorities change so might our knowledge. It is important in any field or application not to loose track of basics we learned and what was important at the start. Although they are rarely used today early 60's, 50's and even WWII forecast techniques in meteorology are still in many cases applicable to modern weather analysis. With expert systems we can now capture and take advantage of all Domain Knowledge, putting it in a structured form which could be useful in providing advice.

3.0 STEPS AND ISSUES IN CREATING AN EXPERT SYSTEM

The evolution of an expert system proceeds through a series of steps or phases designed to improve incrementally the use of the system knowledge.

Step 1 - Knowledge Acquisition and Knowledge Identification

The first step in developing an Artificial Intelligent Expert System (AI/KBES) is the characterization of the domain (in our examples meteorology) knowledge. "Knowledge" itself is the key ingredient to any expert system development. Thus, knowledge acquisition is a crucial aspect of developing an expert system. It is important, not only because knowledge is necessary to make the expert system run, but also because of the importance of expert system developer-user interaction. The term "knowledge engineer" has been associated with the person who collects the knowledge and formulates the presentation schemes.

The knowledge can be embedded in the literature or acquired from experts. Various techniques can be used to acquire knowledge from experts including interviewing or observing the expert at work.

The literature review is an important first step in this process as it sets the stage for later interviews. The review is straight forward with domain and other resources examined.

The knowledge engineer usually has intensive interactions with the domain experts. This poses some interesting situations in many AI applications where AI-oriented knowledge engineers try to become pseudo-experts in a particular field. Sometimes it is successful, many other times it is not.

In the meteorology field, for example, there is no reason why meteorologists cannot become knowledge engineers themselves, provided they have the proper training. A readily apparent analogy can be drawn between computer scientists and meteorologists. Most meteorologists can program in FORTRAN or other common computer languages. They also know how to logically design a structured program. Consequently, they can do much of the work themselves. Should a more theoretical programming problem be encountered, a computer scientist could be called in to help the meteorologist resolve the difficulty. Similarly, in AI-meteorology, meteorologists can be trained as knowledge engineers and conduct interviews and structure expert systems themselves. Should a major difficulty be encountered, the meteorologist could call on a (pure) knowledge engineer or theorist.

Also important is credibility of both the system concept and the knowledge engineer in the eyes of the expert. One method to maintain a good working relationship is to keep the engineer within the field of the expert. By doing this, the knowledge engineer becomes more believable.

We have found these considerations important in our efforts in developing several systems for the Air Force and Army.

STEP 2 Knowledge Representation

This step involves selection of an architecture appropriate for system design. Two general approaches include frames and rules. A frame contains knowledge about a topic in the form of slots; rules contain individual if-then or similar structures. This phase also involves selection of either an AI language such as LISP or PROLOG or a shell, which is a software package that aids in expert system development through input of rules and knowledge much like Lotus 1-2-3 aids in graph creation through data input by the user.

Using these approaches, knowledge can be represented using a variety of techniques. The two major architectural mechanisms for representing knowledge are:

- Forward Chaining
- Backward Chaining.

Under forward chaining, the entire process is data driven and the various rulepaths within the expert system examine the available data and try to test any data-specific hypothesis to acquire more facts or knowledge about a situation. This architecture could be most useful in the emergency threat environmental area (i.e., dispersion of toxic gas) or other areas where final goals are not clear.

Under backward chaining, the rulepaths are oriented toward a common main goal. This goal is achievable if the rules satisfy various subgoals.

The problem today is that many applications have been restricted to one approach. For example, because of our past familiarity with and simplicity of the representation schemes associated with rules, many developers are by-passing newer or more efficient approaches to data representation. One of these approaches is the use of frames. In meteorology, and in particular forecasting, it is necessary to satisfy several subgoals before reaching a conclusion (forecast). Frames provide an excellent method of focusing resources on only subgoals of importance for a particular case, avoiding unnecessary and time-consuming execution of all the rules.

STEP 3 Evaluation and User Acceptance

Technically speaking, the development of an expert system is always being evaluated because the development under each phase is considering questions such as:

- Is the knowledge representation adequate or should it be modified?
- Can users easily interact with the system?
- Are rules consistent with the expert's opinion?

Evaluation of expert systems can be classified into two broad categories: quantitative and qualitative. Quantitative evaluation includes derivation of statistical results from either real-time or past events. This involves compilation of observed versus expert system predicted results. Qualitative analyses is much more difficult, but centers around the fundamental question: "Did the expert system help the user?" If the expert system is able to meet the needs of the user and provide him with advice on recommendations, then the system has fulfilled its job.

Both types of evaluation (qualitative and quantitative) are important, but unfortunately many individuals only emphasize the statistical, quantitative-type evaluation.

Shown in Fig. 2 is a standardized form developed for qualitative evaluation as part of our effort to develop 3 expert systems to predict low visibility at various Air Force bases along the east coast (see Fig. 3).

These systems continue to undergo evaluation. Looking back to some of the earlier qualitative results, the forecasters were very receptive to the computerized expert system forecasting approach to providing them with advice. The user liked the system in 118 out of 143 reported cases. Probably more significant and interesting, however, is that in several cases the forecasters actually decided to amend their forecasts on the basis of assistance they received from the system. In several other cases, forecasters indicated they were alerted to weather factors that they might not otherwise have considered. The point is that qualitative, statistical-type evaluation would never reflect these significant results.

1. Date _____

2. Time _____ (Local)

3. Did you like the way the system interacted with you? (Circle one)

   Yes No
   Why?_____
   _____

4. Did you understand all the questions that it asked you?

   Yes No
   What questions did you not understand? (if any)

   _____
   _____
   _____

5. Would you prefer a different method of entering the information that the system requested?

   Yes No
   Why?_____
   _____
   _____

6. Did you use the system . . . (Check one)

   as needed _____
   just prior to TAF time (1 hour) _____
   other times (specify) _____

7. Are the explanation features adequate?

   Yes No
   If no, why not? _____

8. Do you think this sytem will be helpful in your everyday routine? (Circle one)

   Yes No
   Why?_____
   _____
   _____

Figure 2. User evaluation form

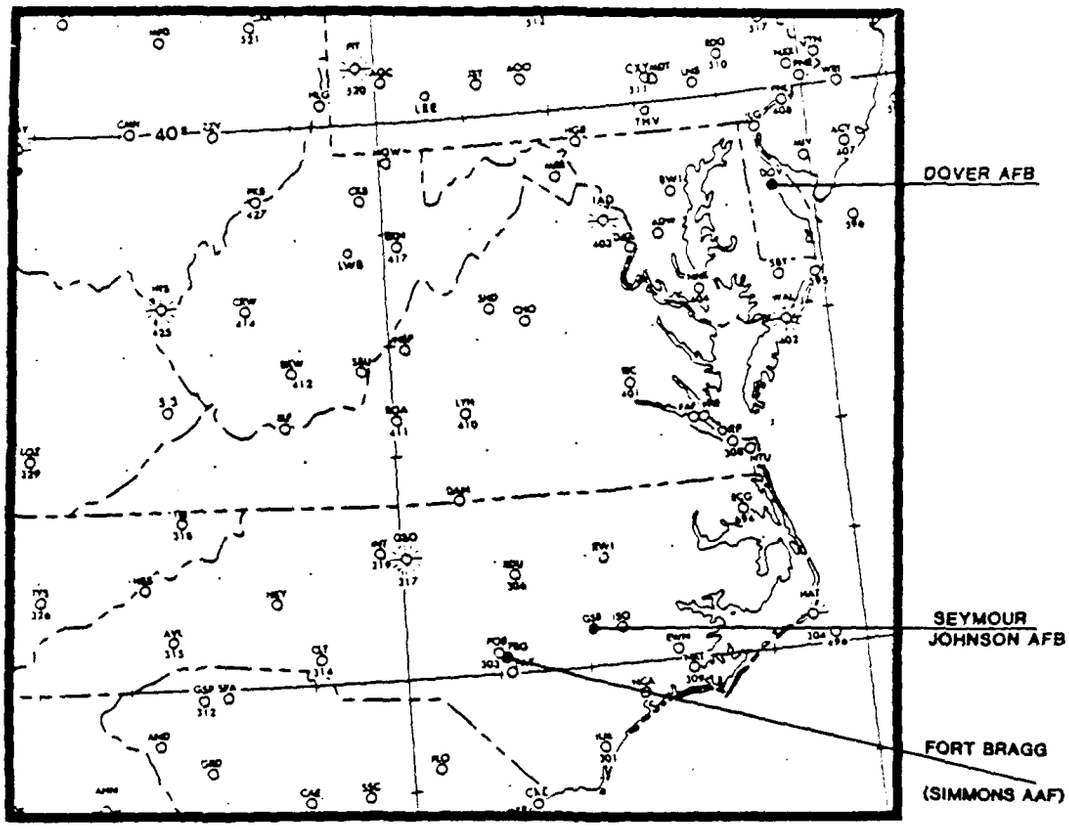


Figure 3. Location of three study air bases.

The survey forms also played an integral role in indicating to the knowledge engineer concerns and problems found with this system. Based on several responses, modifications were made.

When we look at the statistical evaluation, two types of comparisons have been made. The first comparison is based on forecast advice from the system during interactive sessions with the user (forecaster) at the three air bases. The second type of comparison includes running the system on historical data and comparing the forecast with observations. Both of these types of comparisons can be applied to other environmental or military type systems. To summarize results briefly, skill scores indicate the system does exhibit skill in the low visibility forecast. The system showed 39 percent improvement over Seymour Johnson forecasters and 63 percent improvement over Raleigh forecasters.

In a second effort, for the Army's Atmospheric Sciences Laboratory, a low-level wind prediction system (PEGASUS) was developed for low-level operations and in particular, Paradrop operations. Unfortunately, we were not able to begin system evaluation until recently. Evaluation and testing began this past

summer at Fort Huachuca, Arizona. Again, a great deal of effort is being placed on a qualitative type of evaluation. Strong emphasis is being placed on determining whether or not the system heightened forecaster awareness and in doing so brought into consideration forecast elements not originally examined. We hope to have results for both this qualitative evaluation and a statistical evaluation in the next several months.

4.0 CONCLUSIONS

In summary, there is compelling evidence that expert systems are well suited to and of significant benefit in both environmental application and in military field-type operations. The question now is how do we maintain our course and stay on the right track? We just discussed the major steps and some of the issues surrounding the development of expert systems. From this several things are becoming clear. We need to train and use domain (field-type) knowledge engineers. We must stay open minded and flexible as to the type of development packages to use. For example, we cannot become trapped in always using if-then-else type rules. Next we must take a hard look at our evaluation methods and results. We cannot just measure a system's worth on its bottom line conclusion. It is of great importance to understand how the system may have impacted upon the user's everyday routine and thought process. Finally, as the AI field and its applications grow, we must begin to think about management techniques to deal with these and other issues.

A HEURISTIC LOW LEVEL TURBULENCE FORECAST DECISION AID

Martin E. Lee
US Army Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002-5501, USA

Abstract

The US Army Atmospheric Sciences Laboratory (ASL) was tasked to find a practical method for predicting mountain lee wave, mechanical, and thermal turbulence that could adversely affect rotary wing aircraft operations at the National Training Center (NTC), Fort Irwin, California. A literature search was carried out and a review of existing turbulence forecasting techniques was made. Applicable turbulence forecasting heuristics were accumulated into a knowledge base which was organized into a decision tree. The decision tree was designed to draw a forecaster's attention to key factors involoved in the production of mountain lee wave, mechanical, and thermal turbulence at the NTC. The decsion tree then served as a design base for the development of a Low Level Turbulence Forecast Aid (LLTFA) software module. The LLTFA software module automated the decision tree analyses to a reasonable degree by testing various input wind speeds and directions, pressure and temperature gradients, and expected temperatures and dewpoints at selected levels in the atmosphere. The LLTFA software was implemented in Turbo Pascal and can operate on most IBM-PC compatible systems. The LLTFA software performed well in preliminary testing by selectively identifying turbulence mechanisms and intensities within short- range time scales (less than or equal to 24 hours). Therefore, this heuristic LLTFA approach appears to be well suited for further development into an expert system.

1 INTRODUCTION

A literature search centered on turbulence forecasting techniques applicable to the NTC problem was conducted. Selected empirical techniques were extracted from various sources (sources are referenced). These techniques accumulated into a set of heuristic rules (Hayes-Roth, Waterman, and Lenat, 1983) that were then assembled in a more or less logical manner into a knowledge representation scheme (Tanimoto, 1987), which took the form of a Decision Tree (Colquhoun, 1987). This preliminary effort was, therefore, focused on the acquisition and design of an expert system knowledge base for turbulence forecasting at the NTC.

The decision tree served as a knowledge engineering design base that eliminated redundancies of less structured approaches and also reduced the subjectivity in making turbulence forecasts. The decsion process logic of this tree is illustrated in Fig. 1. This process logic then served as a plan for the development of software that would allow a user to make use of the accumulated heuristics.

2 DISCUSSION

The decsion process logic, illustrated in Fig. 1, served as a design base for the development of a Low Level Turbulence Forecast Aid (LLTFA) software module. The LLTFA software module automated the sequencing and accessing of the accumulated heuristics to a reasonable degree. The heuristics functioned within the LLTFA software module by testing various input wind speeds and directions, pressure and temperature gradients, and expected temperatures and dewpoints at selected levels in the atmosphere. The LLTFA software was implemented in Turbo Pascal and can operate on most IBM-PC compatible systems.

The LLTFA algorithm breaks the turbulence problem down into the following categories: mountain lee wave turbulence, mechanical turbulence, and thermal turbulence. Each category is selected in series to carry out a cumulative analysis of the potential for each mechanism to produce turbulence (see Fig. 1). Forecasting accuracy of the algorithm was tested by exercising it with data taken for 20 separate dates from 1981-1987 during which aircraft reported encounters of turbulence in the vicinity of the NTC.

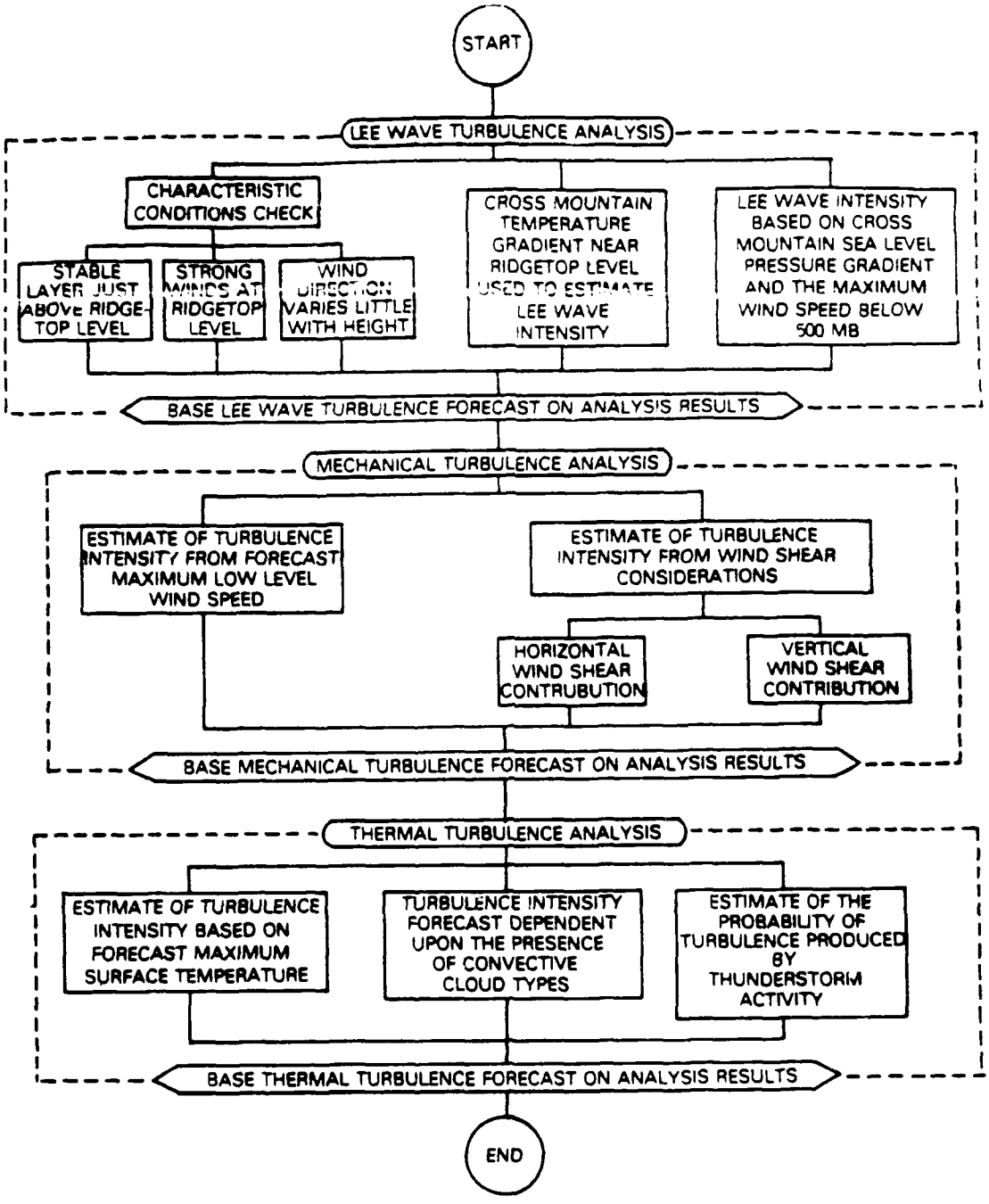


Figure 1: Decision process logic for the Fort Irwin Low Level Turbulence Forecasting Aid in summarized form.

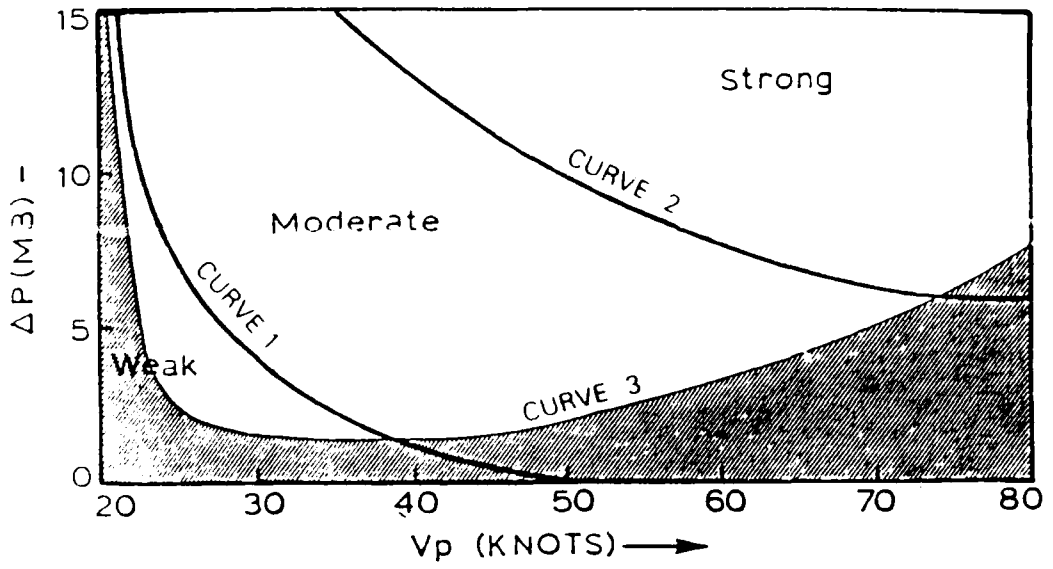


Figure 2: The modified Harrison nomogram above, developed by Lester, 1973 makes use of the relationship between the adjusted large scale cross-mountain sea level pressure gradient and strength of flow aloft. These two parameters are entered into the nomogram to produce an estimate of wave strength. The shaded area indicates that the probability of significant wave occurrence is less than 50 percent.

2.1 LEE WAVE TURBULENCE

Mountain lee waves are smooth waves formed by oscillating air parcels being displaced horizontally over mountain ridges. Complete overturning under some of the waves can create very turbulent, unstable rolls and rotor clouds (Lee, Stull, and Irvine, 1981). Synoptic scale conditions favorable for producing lee waves primarily include a stably stratified troposphere above ridgetop-level, and a wind component greater than 10 mps at the mountain crest, normal to the ridgeline. Wind speed generally increases at a gradual rate well up into the troposphere, but wind direction remains relatively constant with height (less than or equal to 40 degrees change from ridgetop-level to the tropopause in Weather Bureau Technical Memorandum WBTM-FCST-6, 1966). These general conditions are all tested for in the LLTFA (Lee, 1987).

Severe mountain lee wave activity over the NTC has been documented (Banta, 1985). This lee wave activity was attributed to westerly flow over the Sierra

Table 1: 850mb cross-mountain temperature gradient criteria for low level lee wave turbulence intensity prediction. Modified from Lee, Stull, and Irvine (1984) using actual data. Table 1 applies only to the NTC, using Ft. Irwin and Santa Barbara 850mb temperatures. Units are in degrees C per 300 km.

	LIGHT	MODERATE	SEVERE
850mb cross-mountain Temperature Gradient (° C/300 km)	< 3	$3 - 5$	> 5

Nevada and Tehachapi mountains to the west of the NTC. Intensity of these lee wave systems has been related to the large scale sea level pressure gradient across mountain ridges and the strength of the flow aloft (Fingerhut and Lester, 1973). A modified Harrison nomogram, developed by Lester, 1973 makes use of this relationship (see Fig. 2). The curves in this nomogram test were reduced to expressions using a least squares approximation and included in the LLTFA software module for application at the NTC (Lee, 1987). This heuristic method to determine lee wave intensity is actually solved for more objectively by the LLTFA because human errors from manual application of the nomogram are eliminated

An 850mb cross-mountain temperature gradient test from Lee, Stull, and Irvine (1984) is also included in the LLTFA to supplement the nomogram for lee wave turbulence intensity analysis (see Table 1). Table 1 contains temperature gradient test values that were modified specifically for use at the NTC using actual observations of temperature gradient characteristics during turbulent events at the NTC (Lee, 1987). This type of parameter testing, by comparing of observed values to critical threshold values, is characteristic of the rules of thumb used throughout the LLTFA.

2.2 MECHANICAL TURBULENCE

Friction reduces wind velocities at the surface. This can lead to very strong wind shears just above the surface which produces turbulence, and roughness elements such as mountains, buildings, or trees produce wake eddies making it difficult to characterize the low level wind field (Lee, Stull, and Irvine, 1984). It is necessary to consider a low level mechanical turbulence analysis even though mountain lee wave turbulence may be the driving turbulence mechanism aloft (Lee, 1987). This is especially applicable to very low level flight operations (0-

Table 2: Thermal convective intensity criteria based on the maximum surface temperature forecast. Surface temperatures in degrees F. From Novlan (1982).

	LIGHT	LIGHT-MODERATE	MODERATE
Maximum surface temperature forecast (° F)	70 – 79	80 – 89	> 90

300 feet AGL). The LLTFA provides this analysis by examining the availability of kinetic energy, along with horizontal and vertical wind shear tests. These tests are similar to the type of heuristic or empirical rule of thumb employed in Table 1.

2.3 THERMAL TURBULENCE

Turbulent convection is initiated in statically unstable air and is felt near the surface as bumpy thermals of hot, rising air experienced on sunny days. Criteria for predicting this type of turbulence in the LLTFA includes a heuristic test on the forecast maximum surface temperature that is given in Table 2.

Turbulence can also be anticipated in the vicinity of convective clouds according to Novlan (1982). This is also considered in the LLTFA knowledge base via the K-index test (Lee, 1987).

2.4 LLTFA TESTING

Forecasting accuracy of the LLTFA algorithm was tested by exercising it with data taken for 20 separate dates from 1981- 1987, during which aircraft reported encounters of turbulence in the vicinity of the NTC. The results of this testing were all positive, and indicated that lee wave turbulence was a contributing mechanism for 75 percent of all cases considered; 58 percent for mechanical turbulence; and 70 percent for thermal turbulence. Only 54 percent of the thermal turbulence events were the result of surface heating and free convection; the remaining events in this turbulence class represented moist, convective activity, generally produced ahead of deep, approaching troughs. 86 percent of all detectable mechanical turbulence events were associated with lee wave events, which is in agreement with present turbulence model predictions (Lee, 1987).

3 CONCLUSION

Complicated meteorological conditions arise at Fort Irwin, where low level turbulence is known to seriously affect helicopter flight operations. This situation makes it necessary for the forecaster to understand the nature of existing conditions in the vicinity of the NTC before attempting to make a forecast. Even a short-range forecast in this situation requires a forecaster to assimilate large quantities of weather data and to conceptualize a model of the environment before extrapolating this forward in time. This is essentially the approach taken in "nowcasting", according to McGinley (1986). Through actual field deployment at the NTC, it has been shown that the LLTFA knowledge base can provide valuable expertise to assist forecasters in this conceptualization process. Further testing of the LLTFA, carried out using even less data than is characteristically available at the NTC, indicates that the LLTFA guidance was able to selectively identify the significant contributing turbulence mechanisms and intensities within short-range time scales (less than or equal to 24 hours). The LLTFA has thereby proven itself to be a useful tool and practical first step in filling a need at the NTC for low level turbulence prediction guidance. In that sense, the LLTFA presented here is a successful application of primitive Artificial Intelligence (AI) principles within the expert system arena. The heuristic LLTFA approach is, therefore, a promising candidate for further AI development into a more sophisticated expert system, with emphasis given to evolving a reasonable inference engine.

References

Banta, R., 1985: Investigation of O-2A Accident, 27 Mar 85, Air Force Geophysics Laboratory (AFGL), Hanscom Air Force Base, Massachusetts 01731

Colquhoun, J.R., 1987: "A Decision Tree Method of Forecasting Thunderstorms, Severe Thunderstorms and Tornadoes," Bureau of Meteorology, Darlinghurst, 2010, Australia

Fingerhut, W. A. and P. F. Lester, 1973: Lower Turbulent Zones Associated With Mountain Lee Waves. Paper No. 30, NSF Grant GA-32403, Department Of Meteorology, San Jose State University, California 95192, 124 pp.

Hayes-Roth, F., D. A. Waterman, and D. B. Lenat, 1983: *Building Expert Systems*, Addison-Wesley Publishing Company, INC., Don Mills, Ontario, Canada, 444 pp.

Lee, D.R., R.B. Stull, and W.S. Irvine, 1984: Clear Air Turbulence Forecasting Techniques, USAF, AWS, Global Weather Central, Offut AFB, NE 68113

Lee, M. E., 1987: "Low Level Turbulence Forecasting Aid," EOSAEL/TWI Con-

ference Proceedings, U.S. Army Atmospheric Sciences Laboratory, White Sands Missile Range, NM 88002

Lester, P.F., 1973: An Evaluation of a Lee Wave Forecasting Nomogram. Department of Meteorology, San Jose State University, California 95192, 25 pp.

McGinley, J., 1986:"Nowcasting Mesoscale Phenomena," Chapt. 28 in *Mesoscale Meteorology And Forecasting*, American Meteorological Society, Boston, MA.

Novlan, D.J., 1982: "Section IX, Forecasting Turbulence," *Weather Forecast Manual*, ASL, WSMR, NM 88002

Tanimoto, S. L., 1987: *The Elements of Artificial Intelligence*, Computer Science Press, Inc., Rockville, Maryland 20850, 530 pp.

U.S. Department of Commerce/Environmental Science Services Administration, Weather Bureau Technical Memorandum WBTM- FCST-6,"Forecasting Mountain Waves", 1966.

AN EXPERT SYSTEM APPROACH TO ADVISORY WEATHER FORECASTING

Young P. Yee
David J. Novlan

U.S. Army Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002

ABSTRACT

The objective of this Artificial Intelligence related project is to develop advisory weather forecasting techniques using expert systems programming methods. The prototype expert system will be used to predict adverse weather phenomena at White Sands Missile Range, NM. The range supports a variety of field tests which are highly dependent on present and forecasted weather conditions. Critical missions rely heavily on accurate and timely forecasts and, at times, decisions have to be made expeditiously with the available meteorological information at hand. The forecaster must assimilate large quantities of current met data and make a judicious weather prediciton. An expert system can be an excellent aid in presenting relevant data to the forecaster and in checking for fundamental contradictions or errors in judgment.

The initial development will deal with a forecasting aid for thunderstorm prediction. The approach will be to study actual cases of thunderstorm events and document pertinent data and heuristics, rules of thumb, that will be incorporated into a rule based expert system shell. One of the most important goals in this effort is to construct a prototype expert system in which the knowledge base, i.e. the heuristics for WSMR, can be accessible and easily modifiable so that the heuristics for different sites can be used in the knowledge base. The advantages of the AI approach over conventional programming techniques are as follows: the reasoning engine or procedures can be separated from the knowledge base, the rules are English-like, and the coding is easier to modify for rapid prototyping.

1. INTRODUCTION

The development of new expert systems (E.S.) is changing rapidly because of the availability of a number of commercial expert system building tools. A few years ago, expert shells were primarily used on powerful Lisp computers and were in the price range over $50K. Now the emphasizes has shifted to port versions of expert systems onto

less expensive workstations and in particular personal computers (PC). There has been a major thrust by a number of companies to make PC-based expert systems shells. A summary of the major companies and their product line are presented by LoPiccolo (1988) and Gevarter (1987). Figure 1 shows a few commercially available expert system tools or shells. Tools and shells make it possible to develop an expert system in an order of magnitude less time than would be required with the use of LISP or PROLOG. The developer is able to create a working prototype version quickly and to test out a variety of precedures and approaches. With this in mind, it seemed appropriate to develop an advisory weather forecast expert system using a commercially available expert system tool.

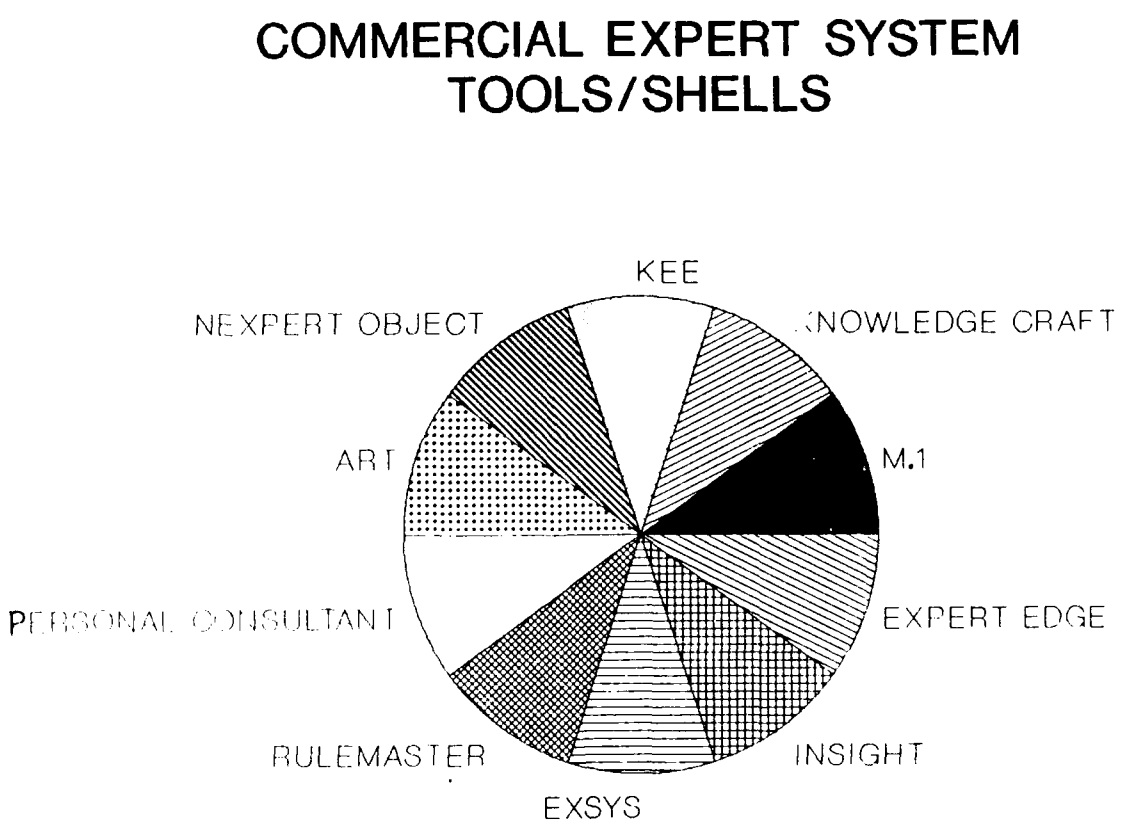


Figure 1. Examples of commercially available expert system tools and shells.

Historically, one of the first successful expert systems to be developed was called MYCIN (Shortliffe, 1976) which was capable of diagnosing blood diseases. MYCIN was shown to out perform even the experts in the field but it was limited in its implementation. Nevertheless, much AI research is directed at solving medical type problems because of MYCIN's early pioneering results. MYCIN is primarily a backward chaining reasoning system based on IF-THEN rules. The PC-based expert system buildirg tool that is being used for the weather forecasting project is somewhat patterned after MYCIN's skeleton structure. The important features of a rule-based E.S. tool for our meteorological applications will be discussed.

2. APPLICATION BACKGROUND

The White Sands Missile Range (WSMR) in New Mexico supports a variety of field tests which are dependent on accurate forecasted weather conditions. Critical missions rely on timely forecasts since decisions have to be made expeditiously with the available meteorological data at hand. The forecaster may be inundated with large quantities of current met data even some that are questionable, and he must assimilate this information and make a judicious weather prediction. Weather forecasting is characterized by complex theories, copious model products, statistical predictors, and huge outputs of meteorological data. To complicate the matter, the forecaster has to deal with both the spatial and temporal variations in the incoming data. Theoretically, the models should provide accurate, fine-grain predictions but in reality, the models are not yet perfectly developed and can only provide synoptic scale guidance. Often, the art of forecasting resides in the expert's knowledge of how the statistical predictors or models are affected by idiosyncratic features of his geographical region. An expert system to incorporate the mesoscale heuristics of a particular locale can be an excellent aid in presenting important information to a forecaster. The forecaster may have overlooked somewhat obvious or essential information simply because of data saturation.

A prototype expert system is currently being developed to forecast thunderstorm activity at WSMR. The goal is to identify the key parameters that are important to thunderstorm development and to codify these parameters into rules. But before initiating the project, we had to evaluate whether this task was appropriate for an expert system approach. The following guidelines were met before we started.

* No direct physical contact is needed to solve the problem i.e. the problem can be described in words.

* No conventional programs were available that specifically addressed this problem.

* The solution involves judgmental reasoning and does not rely solely on extensive calculations or formal analysis.

* A qualified human expert exists and is available.

* There is a potential cost benefit in training expense and especially in mission cost i.e. avoid mission failures due to inaccurate weather forecasts.

In order for an expert system to be successful, the above issues must be considered.

3. IMPORTANT FEATURES OF A RULE-BASED EXPERT SYSTEM TOOL FOR WEATHER FORECAST DEVELOPMENT

There are a multitude of expert system building tools or shells in the market, covering a wide range of features and user friendly interfaces. The field is indeed very confusing because the experts developing an expert system are not familiar with what features are important for his particular application. Caution must be taken in choosing an expert system tool but a certain amount of knowledge acquisition and knowledge representation training is also important (Harmon & King). In our initial development, we have found that certain features in the M.1 expert system tool being used were important for our application. These features are summarized in the following checklist:

* Easy Rule Construction
* Simple Inference or Reasoning Engine
* Automatic Response Menus
* Legal Value Checking
* Panel Mode which allows Multi-Window Debugging Environment
* Access to a Fact Table or an Array via a general rule
* Unknown, Multiple, Uncertain, and Qualitative Answers

The rule construction must not be complicated and the inference engine should be simple and easy to follow. The automatic response menus greatly reduce the development time since we do not have to deal with dimension statements, do-loops, and proper array matching as in conventional scientific languages. The legal value checking is a convenient method of checking whether the entered data is within reasonable limits. If a wrong value is entered, the program will not error out but instead display the legal values that are required and ask the user to interactively reenter the value. The Panel Mode interactively displays the ongoing reasoning of the expert system including the events and conclusions that have been established. The Fact Table or Array is the bulk of the knowledge base in our application i.e. the thunderstorm forecasts itself. Finally, the expert system is capable of explaining the questions, accounting for unknown or multiple answers, and handling uncertainly in the answers. One must evaluate important features for each application.

4. KNOWLEDGE REPRESENTATION FOR THUNDERSTORM FORECASTING

The weather forecast section of the Atmospheric Sciences Laboratory provides WSMR with four basic services: continuous general and special-user weather forecasts, continuous weather observations (surface and aloft) for special users and for climatological purposes; impact prediction for unguided missile launches; and professional meteorological consultation to range users such as High Energy Laser, Space Shuttle, Cruise Missile, Multiple Launch Rocket System (Novlan, 1982). Most of the forecasts and services are specially tailored to the specific user. Certainly the task of forecasting for multiple users and multiple applications is a monumental undertaking. The authors have decided to address thunderstorm forecasting as a stepping stone to a comprehensive expert system for WSMR forecasting advise. The idea is to modularize the essential elements that are important for thunderstorm activity but at the same token these key elements may be common to other activitites such as wind forecasting, temperature forecasting,

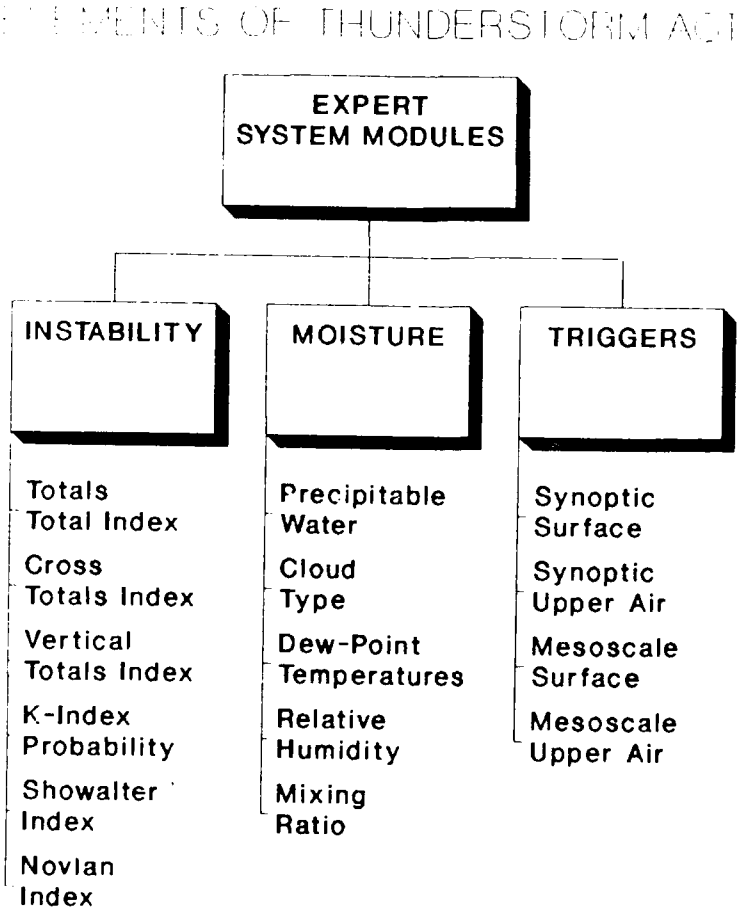


Figure 2. Key elements necessary for thunderstorm activity to occur.

turbulence forecasting, visibility, etc. A decision tree method of forecasting thunderstorms by Colquhoun (1987) describes meteorological parameters considered important for thunderstorms to develop. Elio (1985) outlines representations of storms.

The main ingredients for thunderstorm activity are moisture, instability, and trigger mechanisms as outlined in figure 2. Within this framework there are many heuristics, rules-of thumb, that the forecaster may evoke to help him sort through the myriad of meteorological data that is constantly being received by satellite, radiosondes, numerical weather analyses, NMC national facsimile network, radar, and various teletype networks. We have tried to breakup the analysis into three major areas of concern for convenience and manageability for our expert system but in practice there is constant interaction between these three areas. A prototype E.S. module for evaluation of the instability indices has been developed and the next step will concentrate on the moisture analysis. The most difficult module appears to be the trigger mechanisms since there are temporal and spatial conditions of all

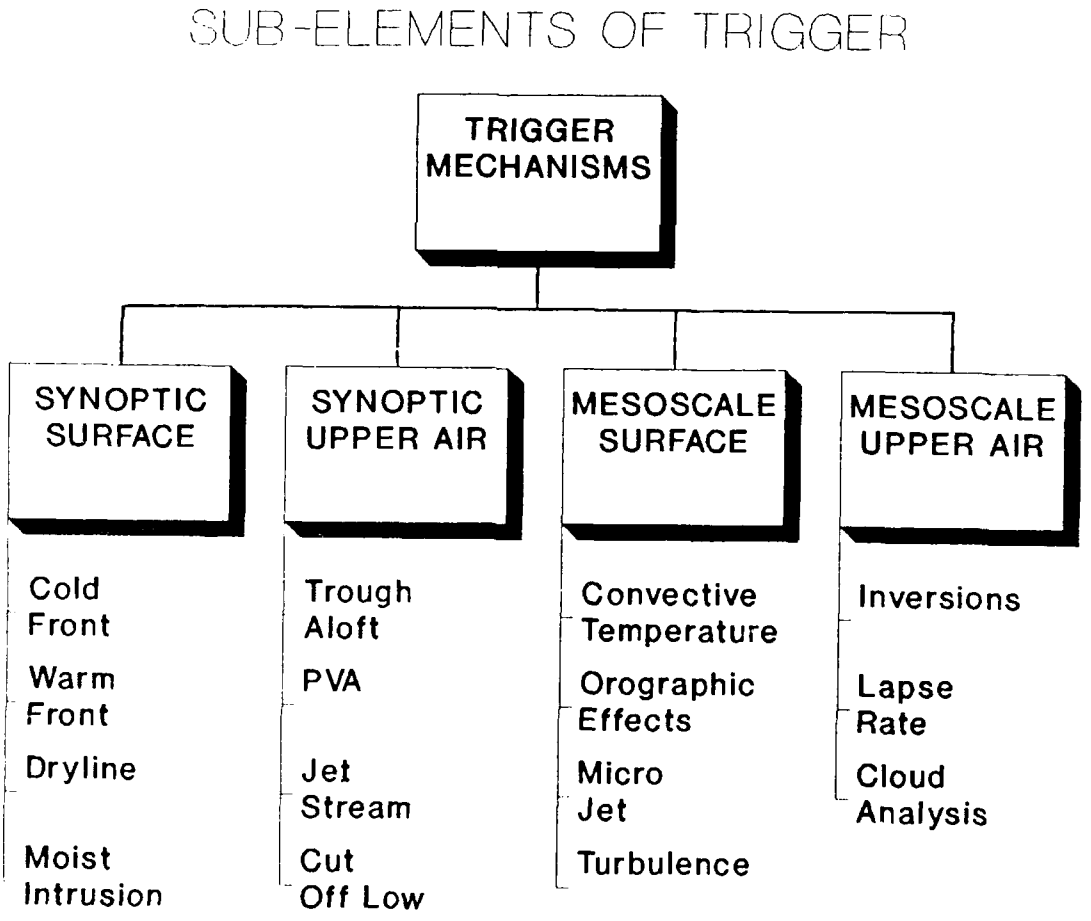


Figure 3. Subelements of trigger mechanisms broken down into four categories for convenience.

scales of motion to consider i.e. synoptic scale and mesoscale, surface and upper air features. Figure 3 gives just a small sample of the exhaustive number of parameters and met features that a forecast will use in his analysis.

5. PLANNED IMPLEMENTATION

The prototyping is currently being done with the use of M 1 expert system tool by Teknowledge, Inc., which is implemented on a PC-AT compatible computer with a minimum of 512K RAM. The source code is written in C language for faster implementation. Since the forecasting problem is too large to handle, we are taking advantage of modularization in building the knowledge base. As mentioned, the process of modularization involves decomposing the large problem into manageable subproblem modules and providing appropriate interactions and linkages between the modules to come up with a cohesive weather forecast. It is hoped that within each defined module we can check the consistency of the input data, eliminate redundant data information, filter out the most reliable data entries, and most important identify the critical parameters and combination of parameters for forecasting an event.

An attempt to program the equivalent effort using BASIC proved difficult to code and especially hard to revise and change the program if the approach was incorrect. Qualitative information was not easily translated into BASIC instructions. Also, the debugging of syntax coding in BASIC took considerably more time and effort than in the E.S. program. The E.S. program is composed of English-like statements that were easy to trace the logical flow of the program although care must be taken to correctly spell the variables and goals

Future plans will have the expert system interface directly with the incoming data as well as receive some numberical results from models. But for now, all the input data is entered manually.

6. SAMPLE TRACE OF THE PROTOTYPE INSTABILITY MODULE

Instability is one of the first critical parameter that a forecaster evaluates for the possibility of thunderstorms. There are a variety of indices that determine instability and the forecaster must screen out the best indicators. For the prototype Instability Module for thunderstorm indices, the user is prompted to enter essential met data to calculate stability indices. If certain data is unknown, a typical default value for that locale is assigned. The module will calculate the vertical totals, cross totals, total totals (Miller,1972) and the K-Value and then assign a qualitative factor to each (low,medium,high or weak,average,strong). From these qualitative factors a search in the knowledge base selects the appropriate thunderstorm analysis output.

```
        ****************************************
        *     THUNDERSTORM INDICES ANALYSIS    *
        *                AT WSMR               *
        *                                      *
        *             Young P. Yee             *
        *            David J. Novlan           *
        ****************************************

Do you want to begin the Thunderstorm Indices Analysis
   (yes, no)?
>YES

What is the month (an integer between 1 and 12)?
>11

What is the day (an integer between 1 and 31)?
>10

Enter the temperature (C) at 850 mb:
>20

Enter the dew point temperature (C) at 850 mb:
>10

Enter the temperature (C) at 500 mb:
>-15

Enter the dew point depression (C) at 700 mb:
>10
------------------------------------------------

Indices forecast is Severe Thunderstorms expected
with peak gusts over 50 knots (TT)

Indices forecast is Scattered Thunderstorms, i.e.
25-45% area coverage, are expected (CT)

Indices forecast is Scattered Thunderstorms are
expected with peak wind gusts about 30-35 knots (VT)

Indices forecast is High Chance that Thunderstorms
will occur today (K Value)

MONTH: 11      DAY: 10
Temperature (850mb)=20      Dew Point Temp (850mb)=10
Temperature (500mb)=-15     Dew Point Depression (700mb)=10

TOTALS TOTAL: 60          TOTALS TOTAL CONDITION: very high
CROSS TOTALS: 25          CROSS TOTALS CONDITION: nigh
VERTICAL TOTALS: 35       VERTICAL TOTALS CONDITION: high
K VALUE: 35               K VALUE CONDITION: high

Final Indices Forecast is Severe Thunderstorms expected
with peak gusts over 30 knots.
```

The preceding printout is an actual consultation with the expert system program for evaluating the instability of the atmosphere. Entries prefixed with '>' are the user's input. The output includes the analysis of the vertical totals (VT), the cross totals (CT), the total totals (TT), and the K-Value (K Value) and finally an analysis from all these indices is presented.

Although this sample expert system appears simple, there are subtleties in the program that may be transparent to the user. For example, embedded in the code are rules that change the threshold values of the stability indices for different months out of the year. Also, we know that stability threshold values for potential convective activity vary widely depending on the location (Western Region Technical Attachment, No.84-14). This factor could be easily included in the expert system. Reasoning such as the following are ideally suited for a backward-chaining, IF-THEN rule based expert system:

> If in the West where most thunderstorms are orographic or air mass in nature, then the vertical totals correlate best with thunderstorm activity.
>
> If West of the Rockies where moisture at 700mb or 500mb is sufficient for thunderstorms, then the cross totals correlate best with thunderstorm activity.

The expert system is not efficient for complex calculations but these calculations can be done by conventional programs and then the results imported into the expert system to evaluate.

7. CONCLUSIONS

In the course of developing an expert system for advisory weather forecasting of thunderstorms, many virtues of using an expert system tool have been discovered as well as certain drawbacks. Although it is easier to incorporate qualitative reasoning into rules, this does not mean that an E.S. tool will solve the problem automatically. Acquiring an expert system tool is just the beginning. A good expert system depends heavily on the expert and knowledge engineer to organize and formulate the rules and exceptions. Especially important is analysis of case studies of actual thunderstorm events to provide the skeleton structure for organizing the knowledge base. It is important to break the problem down into more manageable modules if possible. The thunderstorm project was broken down into three key elements: Instability, Moisture, and Trigger Mechanisms. Results from the prototype stability indices module are very encouraging. We will continue to augment the stability module and then initiate development on the moisture module.

The most important advantage of using an expert system tool for the thunderstorm project is the ability to rapidly prototype a skeleton framework. From this framework we can revise and change the rules

without worrying about the tedious task of debugging the code for syntactical errors. If the framework is constructed properly, augmenting the knowledge base by orders of magnitude should not be difficult. Once the rules and expertise has been fine tuned and established to include only the essential ingredients then adapting the expert system to different locations would only be a matter of changing knowledge bases. In fact, if the procedure for thunderstorm forecasting becomes somewhat well-defined, then the code could be transferred to the more conventional languages. Therefore the deliverable product may not be hardware or software dependent.

Lastly, the authors would like to emphasize the importance of constant interaction between the expert and the expert system because expertise in the field of weather forecasting is a constantly evolving science that needs periodic updating and maintenance.

REFERENCES

Colquhoun, J.R., 1987, "A Decision Tree Method of Forecasting Thunderstorms, Severe Thunderstorm and Tornadoes," Weather and Forecasting, 2, 337-345.

Elio, R. and J. de Haan, 1985, "Knowledge Representation in an Expert Storm Forecasting System," Proceedings of the Ninth International Joint Conference on Artificial Intelligence, August 18-23, 1985, Los Angeles, CA, 400-406.

Gevarter, W.B., 1987, "The Nature and Evaluation of Commercial Expert System Building Tools," Compute, 24-41.

Harmon, P. and D. King, Expert Systems, Wiley Press Book, John Wiley & Sons, Inc., New York, NY

LoPiccolo, P.J., 1988, "Expert-System Shells," Engineering Tools, 64-71.

Miller, R.C., 1972, "Notes on Analysis and Severe-Storm Forecasting Procedures of the Air Force Global Weather Central," AFGWC Technical Report 200.

Novlan, D.J., 1982, "Weather Forecast Manual, White Sands Missile Range," Atmospheric Sciences Laboratory, White Sands Missile Range, NM.

Shortliffe, E.H., 1976, Computer-Based Medical Consultations: MYCIN, Elsevier-North Holland, New York, 1976.

Western Region Technical Attachment, No. 84-14, May 8, 1984, National Weather Service, Salt Lake City, Utah.

SYMBOLIC IMAGE AND TERRAIN PROCESSING TO AUTOMATE THE INTELLIGENCE PREPARATION OF BATTLEFIELD

P. D. Lampru, Jr.
Consultant's Choice, Inc.
Atlanta, GA 30350

ABSTRACT

This paper introduces a fundamentally new idea for automating the Intelligence Preparation of the Battlefield (IPB). The critical concepts are (1) the Symbolic, Synthetic, Software Map (S^3-Map) to represent the terrain and (2) the object-oriented programming paradigm (OOPP) to dynamically simulate the AirLand Battlefield environment. The S^3-Map is a map created by transforming digitized map plates into a novel symbolic representation (i.e., a LISP list), and by autonomously recognizing/labeling the map icons (e.g., roads, deciduous forest, soil types, urban areas, etc.). Once the map icons and other battlefield entities (e.g., military units, weather events, weapons effects, etc.) are represented symbolically, they can be instantiated as objects in the OOPP. In the OOPP, objects can retain the attributes of the entity being modeled; objects can selectively and automatically inherit attribute information; objects can dynamically communicate with other objects by message passing; and objects can autonomously "act" by executing embedded methods or procedures. Thus, if battlefield entities can be realistically modeled as objects, then the dynamic simulation capability in OOPP can support the Terrain, Weather, and Intelligence Analysts by automating many aspects of the Intelligence Preparation of the Battlefield.

I. INTRODUCTION

The Intelligence Preparation of the Battlefield (IPB) is a systematic approach to analyze enemy doctrine, weather, and terrain in a specific geographical area. The purpose of the IPB is to determine and evaluate enemy capabilities, vulnerabilities, and probable courses of action. The IPB process is designed to be highly graphical in nature. It uses military maps, multi-layered overlays, gridded photographs, microform and large-scale map substitutes. The principal components of the IPB process include threat evaluation, weather analysis integrated into the terrain analysis process, terrain analysis integrated into threat analysis, and preparation of products such as the intelligence estimate of the situation.

The IPB is very labor intensive. Because the dynamic nature of the battle makes information perishable, many graphical displays must be produced and a tremendous amount of information must be fused and analyzed rapidly. As the battle evolves, the production of graphics and

subsequent analysis is repeated in a tight decision-making cycle. The critical nature of this process to the battle management, coupled with the need for ever faster decision making, makes the automation of the IPB process a high priority task.

Today, the automation of this process (i.e., the Integrated Meteorological System [IMETS], the Digital Topographic Support System [DTSS], and the All Source Analysis System [ASAS]) focuses on creating and presenting/displaying a wide variety of overlays for interpretation by the analyst. For weather and terrain integration, DTSS is an important first step in automating the terrain analysis process. However, DTSS depends heavily upon a human for all analysis tasks. This reliance upon a human to analyze a wide variety of graphics generated by DTSS may actually overload the analyst, as depicted in Fig. 1. As a result of this condition, shortcuts must be taken in the IPB process so that critical information can be provided to the commander in a timely manner. In other words, automating the generation of terrain displays may not be sufficient to support the IPB process in a dynamic AirLand battle scenario.

In addition, the analysts must perform these highly cognitive tasks under continuous, around-the-clock combat operations. Thus, the quality of the analytic work may be degraded by fatigue, the stress of combat, and the lack of time to adequately perform this work. Therefore, in order to take full advantage of the IPB process, as many analytical tasks as possible must be automated to reduce the workload on the weather, terrain, and intelligence analysts. Figure 2 depicts an automation concept that fully supports the IPB process. This concept depends upon three components--a symbolic representation for the map, a symbolic software development environment, and a parallel processing super-minicomputer. The following section explains why the automation of the IPB process (i.e., specifically weather-terrain integration and intelligent terrain reasoning) depend upon these three components.

2. BATTLE MANAGEMENT ISSUES RELATED TO AUTOMATING THE IPB

Battle management is a fundamental process in the IPB. The map is ubiquitous to all aspects of battle management. Therefore, the single most critical issue to automating any aspect of the IPB process is the computer map representation. It must be capable of supporting robust, intelligent reasoning. Otherwise, attempts to automate the integration of weather and terrain, intelligent terrain reasoning, and combat intelligence information fusion will be very difficult unless the map representation facilitates the implementation of these problems in software.

Digital raster maps, vector maps, and cell-based maps are excellent for graphical display, but we believe they may not be totally adequate in the long run as a foundation to support the software implementation of highly cognitive, analytical tasks that, today, must be performed by a human. We also believe that the application of expert systems to automating these tasks using a digital map representation is a good first step, but that the rule base must eventually be incorporated into a more sophisticated and robust

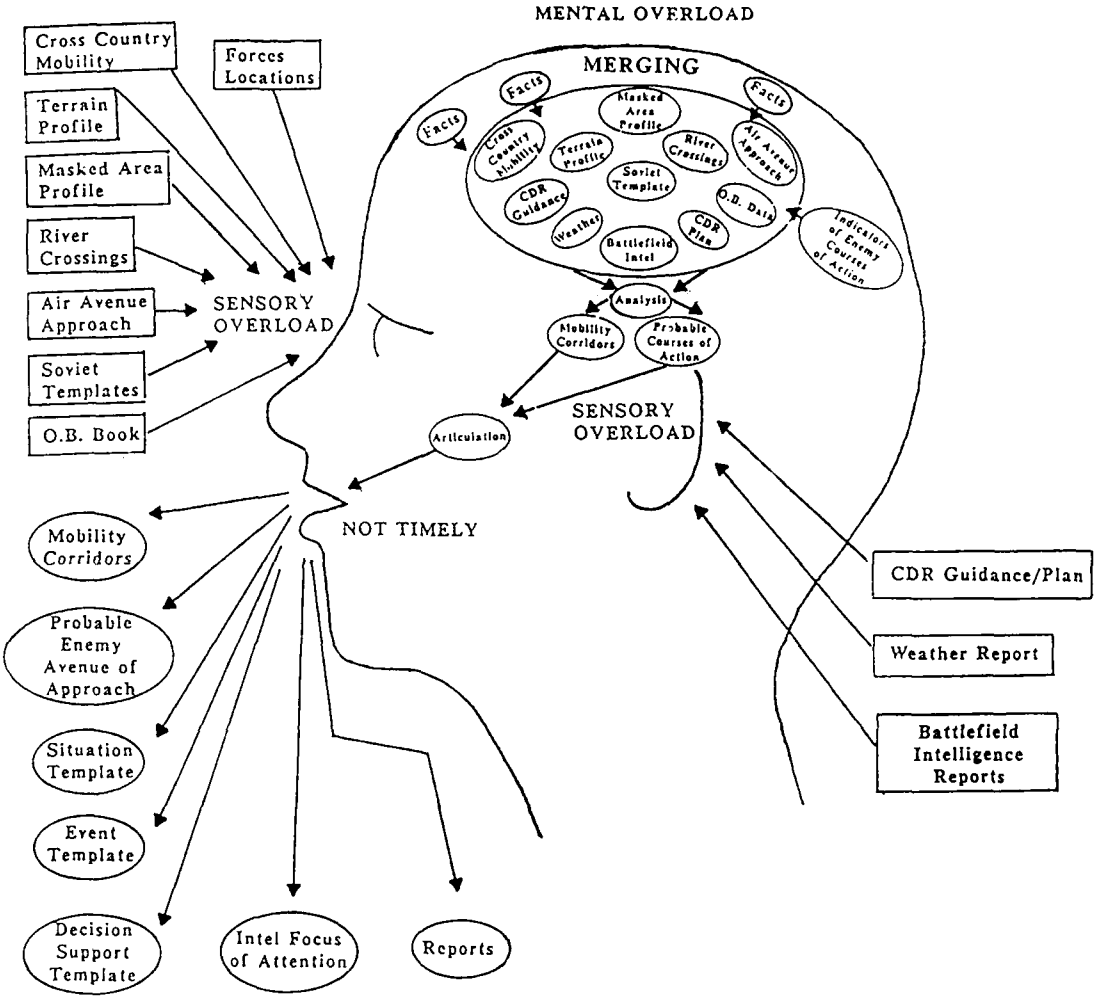


FIGURE 1. The Intelligence Analyst's Nightmare

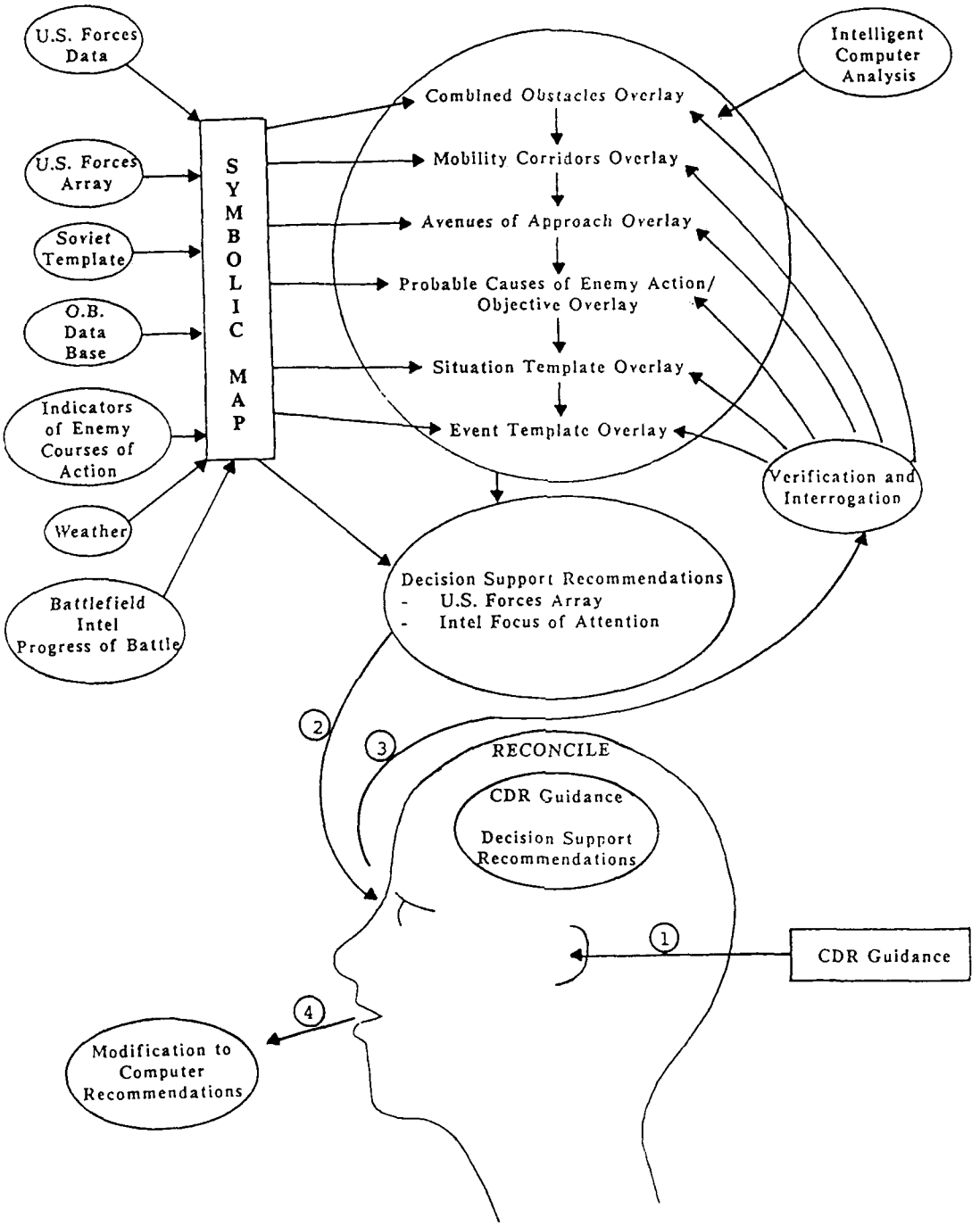


FIGURE 2. Computerized Analysis of Terrain Graphics

inferencing system that can be implemented in a true parallel processing hardware system. In order to truly automate the highly cognitive tasks associated with the IPB process, a software system must be able to perform tasks that simply cannot be handled by an expert system operating on a digital map representation in real-time for the domain. Rather, we believe that the object-oriented programming paradigm, operating on a symbolic map, may have the potential to support the sophisticated inferencing that will be required. There are several problems associated with battle management that are presented to support our beliefs.

First, terrain alteration caused by nuclear, chemical, and conventional weapon systems will drastically change the battlefield's avenues of approach, deny areas to troops, create instant impediments to movement, etc. After the battle begins, the depiction of the terrain on existing maps will not resemble the battlefield. In addition, the effects of inclement weather on the terrain will hinder mobility, channel traffic, and degrade the performance of friendly and enemy weapon systems--the classical problems of weather-terrain integration, weather effects analysis and mobility analysis. Consequently, in order to automate the IPB process, the computerized map representation must be dynamically alterable so that the operator can rapidly change the computer map representation to correctly represent the state-of-the-terrain as it changes over the course of a battle. If all terrain features of military importance can be represented symbolically as objects in the object-oriented programming paradigm, then it may be possible for an operator to interactively and dynamically alter the attribute information of terrain features on the map to accurately represent the state-of-the-terrain. For example, an analyst could instantiate a nuclear explosion as an object at some grid coordinate and cause this object to dynamically alter internal attributes and display information for all terrain feature-objects within a specified proximity to the detonation site.

Second, tactical operations will be totally dynamic because friendly/enemy unit missions, capabilities, and locations will change frequently. Friendly and enemy units must be accurately represented and positioned on the computer map in order to update the map with the current state-of-the-forces. If templating is to be performed, then the computer representation of the units must also be able to interact with terrain features in the map so that their positions can be dynamically adjusted to fit the terrain. When a unit loses or gains capabilities, its resource attributes and combat potential must be rapidly adjusted. A frame-based, or object-based, representation of the military units would allow order-of-battle information to be managed by an analyst and used by the computer to perform an analysis. In addition, military-unit-objects and terrain feature-objects could dynamically interact to template an enemy force's location relative to the map.

Third, the IPB process is a highly cognitive process where inferencing and deducing new information from known facts is critical. A variety of intelligence information must be geographically registered and "fused" before the analysis process can begin. Therefore, the system must be able to intelligently analyze and selectively integrate a variety of data and information into the computerized map representation to maintain the

state-of-the-situation. The object-oriented programming paradigm and the symbolic reasoning capability inherent in LISP should be able to implement these tasks far better than could an expert system operating on a vector or cell-based map.

Fourth, automation of the IPB process should also include the capability to perform iterative simulations to evaluate a variety of scenarios and assess the advantages and disadvantages of different tactical courses of action. Such a capability will require a symbolic representation of the terrain, military units, and the tactical situation; and a programming paradigm that dynamically and realistically simulates the interaction of these symbolic entities. Again, the object-oriented programming paradigm and the symbolic reasoning capability inherent in LISP should be able to implement these tasks.

Fifth, the software needed to automate intelligent terrain reasoning will be extremely large and complex. Therefore, LISP should be the development language of choice, rather than a numeric language like FORTRAN or C. Patrick Henry Winston, Director of MIT's AI Laboratory and coauthor of LISP, stated in an interview, "In my own experience, 200,000 lines of LISP is often more than a million lines of code in another language. I think that in the next ten years, we need to build systems of enormous sophistication in order to handle problems of immense complexity. Those problems will be quite different from anything people have tried to deal with before. They'll involve a heterogeneous mix of subproblems, they'll involve tidal waves of data, they'll require the utmost attention to the real-time assessment of the situations so that the important problems will be solved first, with the less-important problems deferred; they'll require computer-based systems to not only solve problems as an expert would but to act as blunder-stoppers for the kind of decisions a person might make for lack of capability to look at all the data. And to build systems like that will place extraordinarily heavy demands on programmer productivity. And LISP-based programming will be the right way--maybe the only way--to build those systems."

Sixth, an extremely powerful super-minicomputer and a parallel processing environment will be required to support the automation of these functions. The S^3-Map concept implemented in LISP could be deployed in a reasonably priced parallel processing environment based upon transputers. For example, parallel processing transputer boards that transform the Sun workstation into a parallel processing super-minicomputer are readily available (Supercomputing News 1988). Up to eight boards may be used to upgrade a Sun with a total of 32 parallel processors yielding a peak non-vectorized floating-point scalar performance of up to 50 MFlops (Harper 1988). Initial pricing is about $34,000 for a system running parallel common LISP and parallel C with 32M bytes of real memory on a stand-alone workstation (Computerworld 1988). Figure 3 shows how multiple 4 multi-processor units (MPU) boards, each with up to 64 Mbytes of RAM, might be attached to a Sun in a network configuration. These processors would communicate via a parallel path which is separate from Sun's Ethernet (Azzara 1988). Thus, parallel processing technology is commercially available and could be readily adapted for a tactical computer system.

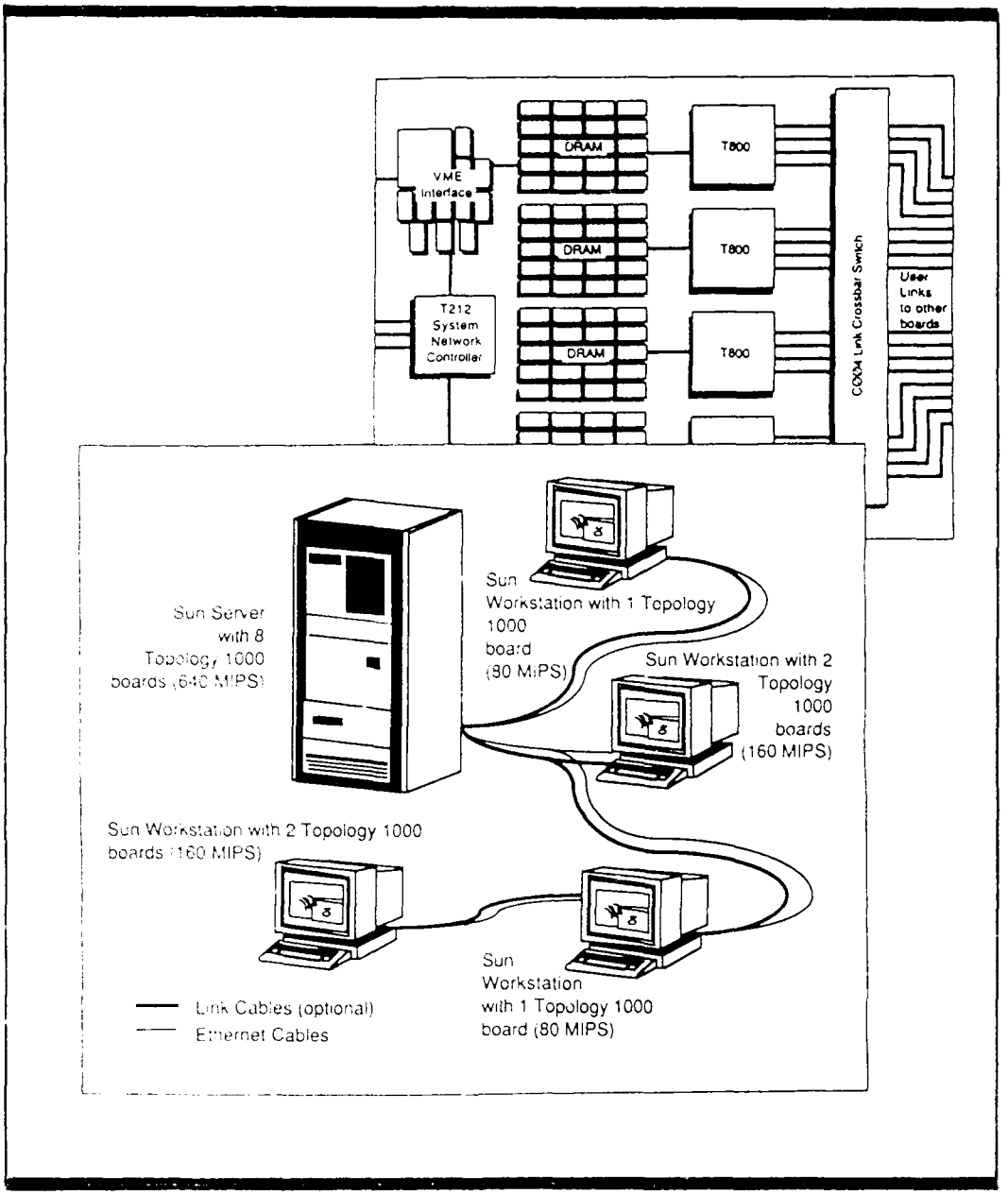


FIGURE 3. Topologix's Topology

Finally, if the IPB process is to be automated, then the most critical issues are (1) the selection of a computer representation for the map and (2) the programming language/paradigm to implement highly cognitive and complex tasks in a parallel processing environment. We believe that the concept of a Symbolic, Synthetic, Software Map (S^3-Map) implemented in an object-oriented programming environment could support the integration of weather and terrain, symbolic information fusion within a geographic information system, intelligent terrain reasoning, dynamic updating of the battlefield situation, and dynamic battlefield simulation.

3. THE SYMBOLIC MAP REPRESENTATION AND SOFTWARE DEVELOPMENT ENVIRONMENT

The selection of a software representation for the map is an issue of prime importance. This representation must be compatible with symbolic languages that can parallel the cognitive process that an analyst uses when visually analyzing the map. Consequently, the approach to performing intelligent terrain analysis should be image understanding and, therefore, the map representation should be symbolic in nature so it is compatible with symbolic languages such as LISP. This approach will enable the full potential of symbolic processing to be brought to bear on the automation of the IPB.

The Symbolic, Synthetic, Software Map (S^3-Map) is a symbolic map that is represented in LISP as a list, rather than as a digital image (i.e., a pixel array or a vector file) (Condon and Lampru 1987). Consequently, the S^3-Map is totally compatible with symbolic languages (e.g., LISP, Smalltalk, etc.). This symbolic map representation instantiates terrain features as objects in the object-oriented programming paradigm. Therefore, the S^3-Map facilitates intelligent terrain analysis (i.e., image understanding by perceptual grouping). The elegance and robustness of this representation for image understanding has been clearly established under a contract from NASA/Goddard Space Flight Center (Condon and Lampru 1988).

The S^3-Map is created through a digitizing, segmentation, transformation, and feature recognition process, as illustrated in Fig. 4. A prototype S^3-Map has been created under IR&D using map separates from a Tactical Terrain Analyst Data Base (TTADB). The creation process begins with a digitized map separate or plate that contains a specific type of map feature. The digitized map is automatically segmented and transformed into a symbolic representation (i.e., a LISP list). This symbolic representation of any image shapes (e.g., terrain features) is actually a width encoded medial axis. Once the map is in this symbolic form, a feature labeling program, that has been told which type of features are on the map, is invoked to label the features and instantiate them as objects in an object-oriented data base management system. An intelligent re-segmentation algorithm then re-connects the disconnected features that extend across multiple map plates and registers all features to a common geographic scale.[1] Thus, the terrain

[1]The image understanding research conducted for NASA/GSFC under Contract No. NAS5-30271 successfully demonstrated in software the ability to autonomously perform intelligent re-segmentation.

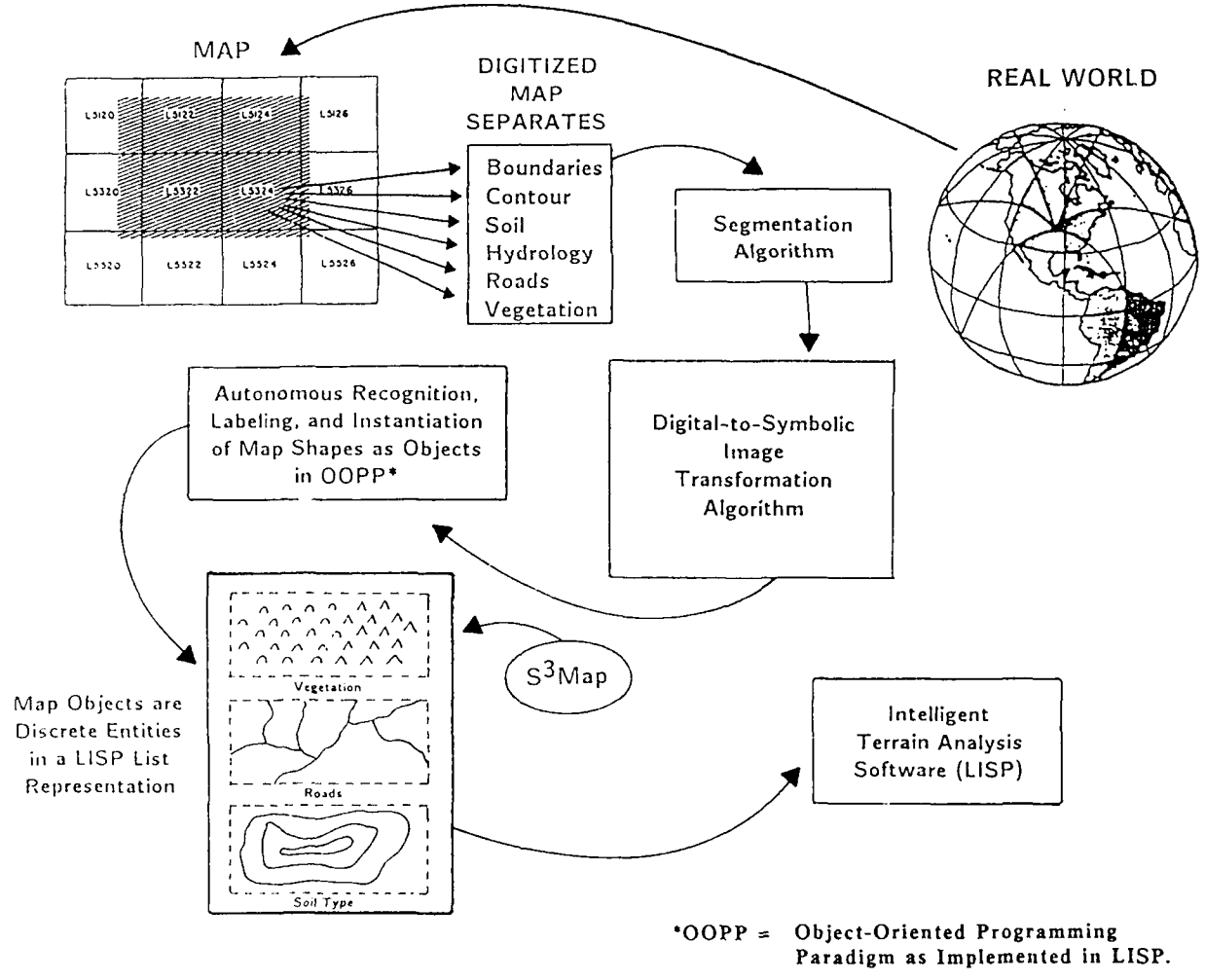


FIGURE 4. Creation of a Symbolic, Synthetic, Software Map (S^3-Map)

features on paper maps are semi-automatically transformed into a symbolic representation that is totally compatible with the symbolic language LISP.

4. WEATHER AND TERRAIN INTEGRATION BY SYMBOLIC FEATURE INFORMATION FUSION

Contract work for NASA/Goddard showed the potential of this representation to perform image understanding by perceptual grouping on Landsat and Seasat images. Contract work for Atmospheric Sciences Lab is ongoing to demonstrate the capability of this representation to autonomously track clouds, and to derive wind vectors and vorticity from multi-temporal GOES imagery. This software development effort should lead to the extraction of synoptic scale features (i.e., fronts, pressure systems, and jet streams), and the recognition of cloud parameters and mesoscale features (i.e., squall lines, thunderstorms, etc.). Ultimately, this image understanding work could provide weather imagery information as input into a weather forecasting model.

Since the original GOES imagery and the output from a weather forecasting process would be in the same symbolic representation as the S^3-Map, it should be possible to perform symbolic-image-feature information fusion and dynamic updating of the state-of-the-terrain. The updating process would be performed by registering the symbolic representation of the terrain and imagery, and causing them to dynamically interact in an object-oriented software environment.

5. CONCLUSION

If an S^3-Map can be constructed from existing map plates, then the military and commercial applications are significant. In the Army, an S^3-Map could "intelligently" assist in the Intelligence Preparation of the Battlefield (IPB), the Estimate of the Situation (ES), and many other tasks for other staff sections. The IPB and the ES include terrain analysis, weather analysis, graphical integration of terrain and weather, identification of avenues of approach, terrain templating of threat forces, war-gaming, identification of named areas of interest, and the posting/displaying of geographically oriented battlefield information or events. The S^3-Map could create terrain analysis products similar to those created by the terrain profile model, masked area model, target acquisition model, perspective plot model, oblique projection model, etc.

In the Corps Tactical Operations Center (CTOC), an S^3-Map could replace the grease pencil and map board. Intelligence information from the G-2's S^3-Map could be provided directly. Friendly forces could be graphically displayed and manipulated. Battlefield information and events could be posted and displayed as desired. Updated personnel and logistics information could be received directly from the G-1 and G-4 S^3-Maps. As the tactical operations center reallocates resources and posts the geographical relocation of units to the S^3-Map, the other staff maps could be updated directly. In summary, the potential uses for a symbolic map and image representation scheme that facilitates "intelligent" processing and symbolic information fusion in a Geographic Information System are considered significant.

REFERENCES

Azzara, M., July 1988: Topologix Seeks to Enhance Sun, UNIX Today!

Board from Topologix Brings Parallel Processing to Workstations, Supercomputing News, 1, No. 2, July 1988.

Computerworld, January 1988: XXII, No. 3, 140.

Condon, M.D., and P.D. Lampru, October 1987: The Symbolic, Synthetic, Software Map, Papers and Proceedings of the 10th Annual Applied Geography Conference, 10, 7-13.

Condon, M.D., and P.D. Lampru, August 1988: Phase I Final Report, Symbolic Imagery Management System, NASA Goddard Space Flight Center, Maryland, Technical Report, Contract No. NAS5-30271.

Harper, J., Summer 1988: Simulation of a 1,024 Processor Parallel Machine, Lucid Moments, 2, No. 3, 1.

Winston, P.H., 1988: An Interview with Patrick Winston, Computer Science and Engineering, Addison-Wesley Publishing Company, 1 Jacob Way, Reading, MA.

ARMY REQUIREMENTS FOR AN INTELLIGENT INTERFACE TO A REAL-TIME METEOROLOGICAL DATABASE

G. McWilliams and S. Kirby
U.S. Army Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002-5501

C. Fields, C. Cavendish, M. Coombs, T. Eskridge,
R. Hartley, H. Pfeiffer, C. Soderlund
Computing Research Laboratory
New Mexico State University
Las Cruces, NM 88003-0001

ABSTRACT

The U.S. Army Atmospheric Sciences Laboratory (ASL) is developing an automated software system to help provide for the optimum utilization of data within its Integrated Meteorological System (IMETS). Optimum utilization is here defined as the timely dissemination of the most representative meteorological data for a specified application. This software system referred to as MERCURY will serve as an intelligent interface between the IMETS database and the tactical users of the IMETS data. Most of MERCURY's design requirements emanate from the fact that it must function autonomously in a tactical environment and have the ability to handle the wide array of meteorological data potentially available from IMETS.

1. INTRODUCTION

The U.S. Army has the need to automate the on-site selection of the optimum meteorological data available for supporting various tactical operations (Coombs, et. al., 1988). Automating this selection process will provide the Army with the ability to more completely utilize the wide array of meteorological data potentially available for tactical operations, to more readily disseminate the data to tactical users, and to reduce the level of manpower required to support this effort. The Army is addressing this need by developing a software system which will serve as an intelligent interface between the Army's Integrated Meteorological System (IMETS) and the tactical users of the IMETS data. The prototype of this software system is called MERCURY, named after the Greek messenger god.

2. BACKGROUND

IMETS is a new tactical system the Army is developing which will collect real-time meteorological data from a variety of sources such as environmental satellites and surface stations and disseminate these data to the tactical users (Harris, 1986). IMETS will also have direct access to a terrain elevation and surface feature database.

The tactical user most often requests meteorological data to run models called tactical decision aids (TDAs) which describe the environmental effects on tactical equipment and operations. These equipment and operations encompass activities ranging from aviation to smoke obscuration.

MERCURY's task of providing the optimum meteorological data requires knowledge not only about the data within the IMETS database but also knowledge about what constitutes the preferred data for a specific user (TDA). This task can be monumental given the large volume and diversity of meteorological data collected by IMETS and the wide range of tactical operations needing meteorological data. Most of MERCURY's operational and design requirements emanate from the fact it must function autonomously in a tactical environment and have the ability to handle the wide array of meteorological data potentially available from IMETS. Examination of the major requirements will show why this interface system needs to be "intelligent" and why it can often use techniques commonly associated with artificial intelligence.

3. OPERATIONAL REQUIREMENTS

Some of MERCURY's most important operational requirements include:

* The ability to process data observed within a mesoscale region approximately 300 kilometers square.

* The ability to quickly adapt to new regions and to the reconfiguration of data sources within a region.

* The ability of knowing the types of meteorological data required by specific applications.

* The ability to use an alternate data source when the preferred data source for a specific application is not available.

* The ability to decide the best method for determining the most representative data for a target area.

* The ability to operate in near-realtime.

The areal dimensions of the Army corps echelon primarily dictates that MERCURY be able to process data for a region 300 kilometers square. It is at this level of organization where the Army assumes responsibility for providing meteorological support for all of its tactical activities. These dimensions also represent the region where terrain and surface feature effects on meteorological conditions become of utmost concern.

The dynamic and mobile nature of the Army's tactical mission necessitates that MERCURY be easily transferable (in the software sense) to new regions and that it be able to quickly adapt to new and frequently repositioned data sources.

The numerous tactical users (TDAs) require various types of meteorological data. Consequently, MERCURY must have knowledge of the preferred data for each of the users. Also, when the preferred data is not available, it must have the ability to derive or find alternate data which are acceptable.

Determining the optimum or most representative data for a specific user application requires a decision as to the method of providing the data. Four such methods are: 1) using data straight from measurements or numerical predictions which are considered representative of the target area, 2) using data from measurements which have been corrected for any terrain, surface feature, and elevation differences between the measurement and target areas, 3) using gridded data obtained from objective analysis of the local data, and 4) using climatology. Combinations of two or more of these methods may be needed in some situations; in these cases MERCURY must be able to determine which methods to use and know how to combine the results.

MERCURY must have the ability to operate in near-realtime since most tactical missions are time critical. Here near-realtime is defined as the ability to dispatch new data to a user every 15 minutes. This operational requirement is primarily driven by the expected frequency at which IMETS will update its database.

4. SYSTEM DESIGN REQUIREMENTS

In order for the operational requirements to be met, MERCURY's system design should be characterized by two very important features. First, the design should embody a data fusion system capable of combining both similar and dissimilar data types. Second, the design should contain a reasoning system for use at critical decision nodes encountered during the interpretation and processing of the data.

The purpose of data fusion is synergism. Fusion techniques provide additional information by the correct juxtaposition or combination of two or more data types. MERCURY will use both homogeneous and heterogeneous modes of data fusion. Homogeneous fusion refers to the combination of similar data types and heterogeneous fusion refers to the combination of dissimilar data types. The requirement for homogeneous fusion is exemplified by the combination of rawinsonde and satellite sounder data to give a more complete thermodynamic profile of the atmosphere. The requirement for heterogeneous fusion is exemplified by techniques which juxtapose terrain, surface feature and meteorological data to determine if a specific measurement is more reflective of either microscale or mesoscale conditions.

This fusion process is predicated upon having a reasoning system. This reasoning system will manage the fusion process in such a way that it can efficiently utilize the fusion information. The reasoning system will be enlisted at critical decision nodes encountered during the interpretation and processing of the data. These decision nodes occur, for instance, when it is necessary for MERCURY to determine the

best method for providing the most representative data or to make a judgement about the effect of terrain and surface features on a particular meteorological data set. The reasoning system has the added challenge of having to operate in a very unstructured task environment which is characteristic of most tactical situations.

5. CONCLUSIONS

The Army's requirements for an intelligent interface to a real-time meteorological database are sandwiched between the fledging technology of meteorological data collection, archiving, and displaying and the advancing technology of the tactical user community. These requirements have been outlined at both the operational and system design level. The system architecture addressing these requirements is described in an accompanying paper (Fields, et.al., this volume).

6. REFERENCES

Coombs, M., C. Fields, and G. McWilliams, 1988: Artificial Intelligence Methods for Optimizing the Use of Meteorological Databases: Recommendations for Implementing the MERCURY System. Technical Report ASL-CR-88-0034-2, U.S. Army Atmospheric Sciences Laboratory.

Harris Corporation, 1986: Integrated Meteorological System (IMETS) Interface Control Document. Harris Corporation Government Information Systems Division, Melbourne, Florida.

CONCEPT FOR WEATHER RELATED DECISION AIDS FOR THE TACTICAL COMMANDER

Bernard F. Engebos, Robert R. Lee, and Robert L. Scheinhartz
U.S. Army Atmospheric Sciences Laboratory
White Sands Missile Range, New Mexico 88002-5501, United States

ABSTRACT

Terrain and weather affect combat more significantly than any other physical factors. They provide opportunities and impose limitations, giving a decisive edge to the commander who uses them best. Although commanders have virtually no control over weather, they can take advantage of it or at least minimize its effects through the use of weather related decision aids. Over the past several years, the U.S. Army Atmospheric Sciences Laboratory (ASL) has developed weather related decision aids for various Army automated systems. A short history of ASL's developments in this area will be presented along with typical examples. Future plans for generating weather intelligence decision aids, emphasizing expert system techniques, will also be presented.

1. INTRODUCTION

Weather support to the U.S. Army historically has been divided between meteorology for intelligence and that for Field Artillery support. This paper addresses that portion dedicated for intelligence purposes only. Intelligence Preparation of the Battlefield (IPB) is a procedure for integrating data on weather, enemy, and terrain to support the planning and execution of combat operations. This is especially needed today for the Airland Battlefield with its highly sophisticated weapon systems. This IPB process is performed at division, corps, and echelons above corps.

The air-land battlefield is an extended, integrated battle involving the use of all air and land forces. Conventional, nuclear, chemical, and electronic weapons and counter-measures are integrated to attack the enemy throughout the depth of his formations. Wherever called upon, the Army must be prepared to fight and win by using all available combat power throughout every dimension of the battlefield.

Combat power depends on much more than troops and weapon systems. It depends on logistics, communications, intelligence, security, and a number of other types of combat support and combat support services. Any of these factors may prove to be decisive in the battle and commanders can manipulate and control them to achieve a tactical advantage.

Weather is the single potentially decisive factor over which the commander has little or no control. Yet, the weather may be the most significant factor to be considered. Weather related conditions affect mobility, the requirement for thoroughly integrated air and ground operations, and the ability to see and attack deep. Furthermore, favorable weather frequently enhances the accuracy and effectiveness of complex weapon and support systems. Weather affects virtually every operation, every piece of equipment, and every person on the battlefield. FM 100-5 states that "Weather and terrain have more impact on battle than any other physical factor, including weapons, equipment, or supplies."

History provides evidence of numerous battles won or lost because of the influence of weather. Examples of battles where weather was a decisive factor are the Spanish Armada, Operation Overlord, the Battle of the Bulge, and Napoleon's and Hitler's attempts to take Moscow.

Although commanders have virtually no control over the weather, they can take advantage of it or at least minimize its effects through the use of weather tactical decision aids in the course of their planning.

2. DEVELOPMENT OF WEATHER RELATED DECISION AIDS

During September 1982, the Target Analysis and Planning (TAP) System was developed and implemented on an Apple II+microcomputer in PASCAL. This system was designed primarily for nuclear targeting planning purposes and it was to play a significant role in terms of tactical weather intelligence (TWI) software developments. ASL used this computer system to initiate developments of its TWI software.

2.1 MICROFIX

During December 1983, U.S. Forces Command (FORSCOM) produced its first MICROFIX System Version 1.0. These were based on Apple II software. These in turn, were upgraded in January 1985 to Version 1.2 and to Version 2.0 during February 1985.

ASL developed a series of TWI software packages which directly paralleled these upgrades. These developments were closely coordinated with the terrain analyst and the Staff Weather and Chemical Officers at the 9th Infantry Division at Fort Lewis, Washington. ASL software was provided for integration into the Topographic Work Station Software to personnel of the U.S. Military Academy Department of Geography and Computer Sciences, and to FORSCOM. The latest version of this software, Version 2.25, is scheduled for release shortly.

2.2 MCS-ANBACIS

In 1984, ASL participated in the contract initiation for the Maneuver Control System-Automated NBC Information System (MCS-ANBACIS). This contract was funded by the Defense Nuclear Agency

(DNA) in conjunction with the U.S. Army Chemical School and the 9th Infantry Division. ASL developed software is targeted to play a significant role in this software development as it proceeds to be integrated into the MCS software.

At the same time, DNA requested that ASL develop 3D no-fly boxes for chemical hazard and nuclear fallout predictions. These products are the first NBC products directly aimed to support Army aviators. These products will be incorporated into future versions of the MCS-ANBACIS software.

2.3 SUPPORT TO EXERCISES

During 1984, ASL was invited to participate in two exercises with their APPLE II software in conjunction with commercial and AN/TMQ-30 meteorological surface sensors with personnel of the 9th Infantry Division. Software provided as part of these exercises included the automation of NBC reports and messages, smoke munition expenditure models, effects on personnel in chemical protective clothing, weather data analyses, and critical value weather checklists. This resulted in positive feedback from the troops in the field who used this software.

During 1985, ASL participated in the Field Training Exercise, "Border Star" at Fort Bliss, Texas. This involved a battle between the 9th Infantry Division (Light Division) against an Armored Cavalry Regiment. The MICROFIX Version 1.0 software and the AN/TMQ-30 and fabricated surface meteorological sensors were involved and played a significant role in the battle outcome.

During March 1986, ASL participated in the Army Test and Evaluation Program exercise with the 17th Field Artillery Brigade in Germany with both surface sensors and its TWI software. It was very well received.

3. ONGOING EFFORTS

In 1984, Congress directed that a Test Weather System be developed. ASL is currently working on this software/hardware package with a demonstration of capability scheduled during 1989. This involves both satellite and land based data.

In December 1986, the Operations and Organizational (O&O) Plan for an Integrated Meteorological System (IMETS) was approved. It would be a tactically mobile, automated system to collect, process, and disseminate weather information during combat.

The IMETS mission would be three-fold: weather data collection; preparation of weather products to include forecaster and tactical decision aids; and dissemination of weather products.

The concept of the AirLand Battlefield Environment (ALBE) program was conceived in 1982 by the Corps of Engineers. This also played a

significant role in ASL's weather intelligence efforts. A series of field demonstrations, appraisals, and customer tests have been conducted over the past several years. Plans include several follow-on tests. The MILVAX II computer system is being used for these tests. Both Corps of Engineers laboratories and ASL have successfully participated in these tests.

All IMETS developments have been assigned to the Program Manager for Joint Tactical Fusion. Currently, approximately 160 weather related decision aids are available. New weather decision aids will be developed and exercised on both the ALBE Testbed and the Test Weather System with major field demonstrations of these capabilities planned for 1989. Future ALBE demonstrations are also planned for 1990 and beyond.

4. ARTIFICIAL INTELLIGENCE EXPERT SYSTEMS APPLICATIONS

In order to develop the Test Weather System (TWS) and IMETS software, considerable amounts of data must be ingested and analyzed, and many complex weather related decision aids developed in a near real-time mode. This necessitates the reduction in computer running time and tailoring of existing techniques, analyses, and models. Here is a fertile area for expert system developments.

Examples of expert systems being considered for implementation into IMETS and the TWS and ALBE demonstrations software include:

-A knowledge based system to check the internal consistency of various meteorological parameters, to include data received from satellites and remote sensors, so that bad data are not used to generate decision aids or forecasts.

-The design of expert techniques to automate a system which will select the most useful weather data from available sources, and use it to generate Tactical Decision Aids.

-Expert systems to forecast wind shear and turbulence effects on Army aviation operations.

-Expert systems to forecast probability of precipitation and thunderstorm severity on the battlefield. Neural network theory could be used here.

-Expert systems to be used as frontend and backend analyses scheme for weather related efforts on large and complicated Army systems.

TIES TO THE FUTURE

Bob O. Benn

Assistant Director of Military Programs
Directorate for Research and Development

U.S. Army Corps of Engineers

Our success on the battlefields of the past has been made possible in part by the many products of our research and development programs. But the battlefield of tomorrow will be more complex with more data and information available to the commander than ever before. If we are to continue to succeed we must continue to use similar research and development results in a well integrated and complementary fashion, taking full advantage of the potential of artificial intelligence, robotics and other data management capabilities.

This paper will review some of the technologies we plan to use that highlight the continuing need for advanced applications of artificial intelligence to satisfy the tactical commanders' requirements for knowledge of the battlefield.

Yesterday, today and tomorrow: these are three words that we see very often, but what do they have in common and what do they mean to us here today? Let's look at them in terms of how we fought yesterday, how we fight today and how we might fight in the future.

The trench warfare of World War I now seems bizarre, but it illustrates the technology of the times. Trenches were dug by hand, repeating rifles and slow firing machine guns were "hightech" infantry weapons, and ground mobility was a function of "real" horse power and the horses ability to pull wagons through the mud.

The warfare of today features mobility of fighting forces, lethality of weaponry and maximum use of technology. Smart weapons using electro--optical sensors or infrared heat seekers give the soldier far more flexibility and fire power. Mobility has improved and commanders can more rapidly move their forces where they are needed. However, in many cases the commander is saturated with data and information about the battlefield without the capability to fully exploit this knowledge. He is still making many of his decisions in the same old ways with little assistance from such technological advances as artificial intelligence.

The battle of 2018, only 100 years after World War I, may feature large space structures, sophisticated composite materials, and the commanders' command and control process will be based on extensive use of the type of artificial intelligence and robotics technologies being developed today. Space and the use of the integrated space-airground-based systems will play a major role in supporting the tactical commander in the 21st century. While there are many keys to success on the battlefield of yesterday and today we must tie them together to ensure that we continue to succeed on future battlefields.

Technology, intelligence and environment are the three keys that I feel are most important to our past successes. We have continued to push technology to the limit, used our intelligence to know the enemy and know how best to exploit the technology we developed, and we have tried to make our systems less dependent on or impacted by the environment in which they operate. Each of these three keys have and will continue to be influenced and advanced by the RDT&E community. We will do this because we have customers who rely on our efforts to improve these three key areas.

The battlefield commanders' requirements are being satisfied. To win on any battlefield, the commander needs many things. Among them he must be able to see the enemy at extremely deep range and be able to defeat the enemy in their own rear areas, before they engage the commander's own troops. The commander also needs to assimilate vast amounts of information about his own troops, the enemy, and the environment. Helping the commander to assimilate and fully exploit all of the vast knowledge available about the battlefield is one of the key issues of this symposium. How can we as scientists and developers help to satisfy these basic requirements of the commander?

We can continue to challenge science and exploit all of the technologies that are developed. Our ability to effectively harness the large volume of data that all of our new emerging sensors are developing and turn that data into useful, meaningful information for the commander depends to a great extent on the application of artificial intelligence techniques being discussed at this symposium. In addition, we also need to learn how to control the effects of the environment on future systems or at least identify the effects or impacts the environment will have on the employment of future weapon systems.

During the recent Army Science Conference held at TRADOC, General Thurman identified several

key technologies as needing further exploitation (See Table 1). Our task is to develop these key emerging technologies and help the field commander exploit them through merging and integrating them with the advances being made in artificial intelligence techniques.

General Thurman's challenge to the scientific community is that just developing and exploiting emerging technologies isn't enough. We must learn to leverage our technologies -- get the most return for our dollar -- by using competitive strategies to help us selectively leverage our emerging technologies. One way to achieve this is to develop competitive strategies using the six steps outlined on the left of Table 2.

For illustrative purposes, let us just cite one example of how this competitive strategy might be applied. We first must identify areas that are important or key to the enemy and determine what his strengths are in this key area -- massing of his armor. Then by identifying weaknesses in this strength, we must develop strategies to exploit these

Emerging Technologies		
• Advanced Sensors	• AI/Advanced Signal Processing	• Power Generation/ Storage/Conditioning
• Brilliant Munitions	• Advanced Materials	• Space
• Directed Energy	• Photonics	• Biotechnologies
• Acoustic Devices	• Propellants	• Low Observables
• Chemical Aerosols	• Robotics	

Table 1. Emerging Technologies Identified by General Thurman at Army Science Conference

1. Enemy High Driver within Mission Area	→	Armor
2. Enemy Strengths within Mission Area	→	Mass
3. Weakness within Strengths	→	Limited to Ground Mobility
4. Develop Strategies to Exploit Weakness	→	Alter the Soil
5. Anticipate Likely Enemy Countermeasures (CM)	→	Modify Vehicle Suspension
6. Pre-plan U.S. Counter-Countermeasures (CCM)	→	???

Table 2. Example of Leveraging Technology with Competitive Strategies

weaknesses -- such as knowing the location of weak soil or altering the soil characteristics to limit his mobility. Then, we must be ready to develop counter-countermeasures to the countermeasures we envision him taking in response to our original actions.

A future system based on emerging technologies is the Battlefield Exploitation System (BES) being researched by the Corps of Engineers. BES will make extensive use of artificial intelligence and/or expert system technologies as well as other advances such as 3-D computer image generation to support tactical commanders at all echelons. Tactical decision aids and tailored analyses of the effects of terrain, weather and man-made contaminants (such as smoke and NBC) on tactical operations and weapon systems will be available for battlefield display down to the lowest required level.

The system just discussed is considered future or notional because it is a concept that is in the process of being defined or formulated. A follow-on idea is to integrate several of these knowledge based future systems to make maximum use of their potential to provide data directly to the commander. To achieve this, an integrating system, heavily dependent on artificial intelligence, is being investigated. This new integrating system is called the Knowledge of the Battlefield-Future System or KNOB-FS (or simply KNOBS).

KNOBS will be an integrating system that can provide the link between the commanders' requirements processed in the Concept Based Requirements System (CBRS) and the emerging technologies being developed in various laboratories. The KNOBS will be based on the Airland Battle Future and the army 21 concepts being developed by TRADOC.

Satisfying the many and varied needs of commanders, at all organizational levels, for knowledge of the battlefield in the 21st century may require several specialized as well as generalized systems. Therefore, to ensure that all of theses future systems can be integrated and interfaced, an umbrella future system is needed. KNOBS encompasses several component future systems that are synergistically linked to work in synchronization to collect, process, integrate, and disseminate military environmental information in near real time through the use of space, air and ground sensor systems.

KNOBS is not intended to be a separate entity and will not compete with other command and control systems. KNOBS will operate as part of the Army Command and Control System (ACCS), or its follow-on systems, and ensure that proper support, in the form of tactical decision aids dealing with such areas as weather, terrain, and position/navigation is provided to all functional areas in the battlefield operating system. Extensive use of AI in KNOBS makes it possible to complete the following actions almost simultaneously, and provide these diverse products to users at all echelons.

Tactical decision aid products are produced and provided to the friendly tank forces on flat plasma screens outlining the fastest route to contact with the enemy using a trafficability forecast based on the effect of recent rains on the terrain.

The user on the hill receives an automatic override audio alarm on his helmet radio warning him of the danger from CB attack.

The helicopter pilot is able to select, arm and fire the correct missile toward the proper unseen coordinates based on guidance received from the sensor and expert system on satellite S2 and approval from the command post in the rear.

The commander and staff at the command post receive TDA products that display the overall battlefield situation using 4-D holographic technologies. Products given to the commander include completed intelligence analyses and evaluated options for decision-making, not raw data, and on-site battle action simulation.

In addition to giving us a brief glimpse of how KNOBS will provide support on the battlefield of the 21st century, the previous examples also illustrate why the challenges to the Army Science Board are so important. These challenges include artificial intelligence, networking of tactical level sensors, Army software managements, training simulations and simulators, and modeling.

To succeed in the future and satisfy the requirements of the tactical commanders we must meet the goals and objectives presented in this paper. I have tried to emphasize the fact that technology, intelligence, and the environment have been keys to our ability to satisfy the tactical commanders' needs. We need to continue our efforts along these same lines.

The KNOBS concept being investigated by the Corps of Engineers will provide the mechanism to integrate the emerging technologies and future systems and their numerous system and event data inputs into a readily useable battlefield decision support system. This will give us a single, integrated knowledge of the battlefield system on which we can make decisions to leverage emerging technologies.

Artificial intelligence is important to the full exploitation of essentially all of the future systems now being considered. KNOBS, which will provide the vehicle for the integration of these varied future systems, will also depend on significant advances in artificial intelligence technologies to ensure that all of the required support is provided to the battlefield commanders of the 21st century.

The efforts of the participants of this symposium will ensure that the vast amounts of data that will be gathered about tomorrow's battlefield will become the knowledge needed and used by the commander.

ALLOCATING SENSOR ENVELOPE PATTERNS TO A MAP PARTITIONED BY TERRITORIAL CONTOURS

T. M. Cronin
US Army CECOM Center for Signals Warfare
Warrenton, VA 22186

ABSTRACT

An algorithm is presented which assigns a set of sensor envelopes to a set of territories, based upon minimizing a quadratic objective function subject to a set of quadratic constraints. The objective function is formed by differencing a territorial integral with the integral of the coverage envelope. This difference is called the vulnerability measure. A territorial integral is intuitively defined to be the area delimited by the boundary contour of a territory. A coverage envelope is the intersection of the two-dimensional range pattern of a sensor with a territorial boundary contour. The constraints specify bounds on sensor overkill and redundancy. It is assumed that there are m sensors and n territories to be covered, with $m < n$. Therefore, there are m quadratic equations to be solved, in n unknowns. The solution consists of a mapping which minimizes the total territory not covered by the sensors. The algorithm accomodates coverage patterns of general complexity; e.g., neither the territorial boundaries nor the sensor range patterns need be convex or star-shaped. As utility functions to support sensor optimization, two new algorithmic techniques have been developed. The first is a linear time algorithm to compute the intersection of two simple digital contours of general complexity. A sense of handedness is imparted to the computer due to an algorithm which guarantees counterclockwise orientation of a boundary. This convention assures that the inside is always to the left during a traversal, which implies that the arcs encompassing the intersection can be collected by turning left at crossing areas. The second algorithm computes the integral of a digital contour, and the query time complexity is shown to be linear, directly proportional to the number of unique abscissas in the boundary.

1. INTRODUCTION

Sensor resource allocation is a longstanding problem in nonlinear mathematical programming. The problem may be posed as follows: given a set of sensors and a set of territories, allocate the sensors to the territories in such a way that the area of coverage is maximized, while at the same time controlling sensor overkill and redundancy. This paper addresses three essential issues associated with automated sensor resource allocation: intersecting an arbitrary pair of digitized contours which may possess boundaries of general complexity; computing the integral of the intersection of two such contours where the intersection may consist of multiple pieces which may be multiply-connected; maximizing the sum of the integral of the intersection across all sensor-territory boundary combinations. This paper *does not address* the problem of relieving sensor overload within a territory, nor does it address the problem of ensuring that a particular territory is available for sensor emplacement.

1.1 TERMINOLOGY: TERRITORIES AND SENSORS

Let T' be a topographical region. The *emplacement set* of T', denoted e(T'), is the set of all local maxima of T'.

A *territory* T is a closed contour formed by projecting the boundary of topographical region T' onto the plane from above. We will differentiate between the actual name of a contour T and its linked list β of boundary coordinates.

The *ith emplacement-constrained territorial boundary*, denoted $e_i(\beta)$, is the set formed by uniting the boundary β of territory T with the boundaries of the sets of interior points of T *invisible* from emplacement element $e_i(T')$.

A *sensor* P is a simply-connected, regular closed contour with boundary ρ. The boundary ρ is also called the *sensor envelope pattern* of P. The *centroid* of P, denoted cen(P), is the mean of the boundary ρ. The *radius* of P, denoted rad(P), is defined to be the maximal distance from cen(P) to ρ.

Let P be a sensor. The *ith position of sensor P with respect to territory T*, denoted $pos_i(PIT)$, is defined to be that emplacement element $e_i(T')$ of topographical region T' upon which cen(P) is situated.

1.1.1 The Coverage Envelope

Let $e_i(\beta)$ be the ith emplacement-constrained boundary of territory T, and ρ the boundary of sensor P. Then the *coverage envelope* $\xi_{i(\beta)\rho}$ is defined to be:

$$\xi_{i(\beta)\rho} = e_i(\beta) \cap \rho \tag{1.1}$$

1.1.2 The Vulnerability Measure

Let $e_i(\beta)$ and ρ be defined as above. Then the *vulnerability measure* $V_{i(\beta)\rho}$ is defined to be:

$$V_{i(\beta)\rho} = \int e_i(\beta) - \int \xi_{i(\beta)\rho} \tag{1.2}$$

1.1.3 The Wasted-Resource Measure

Let $e_i(\beta)$ and ρ be defined as above. Then the *wasted-resource measure* $W_{i(\beta)\rho}$ is defined to be:

$$W_{i(\beta)\rho} = \int \rho - \int \xi_{i(\beta)\rho} \tag{1.3}$$

Figure 1 graphically portrays the concepts of vulnerability and wasted-resource.

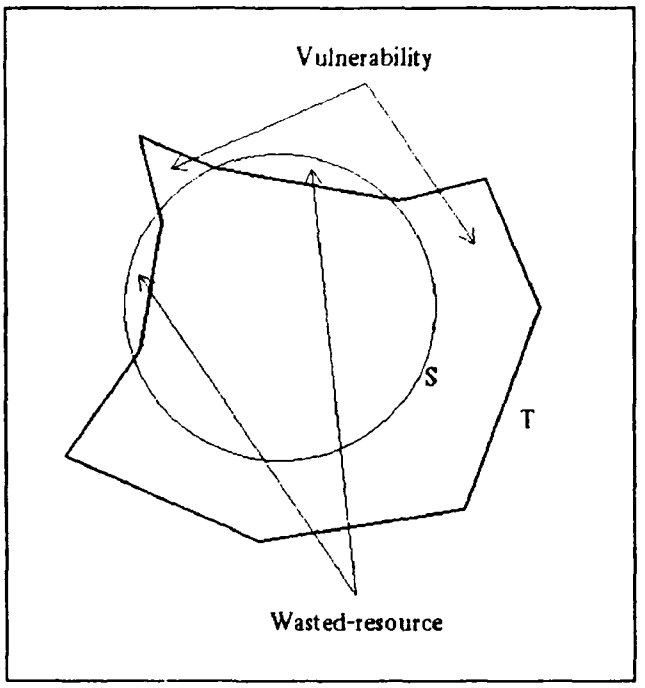


Figure 1. A graphic depicting the concepts of vulnerability and wasted-resource. Boundary T is a territorial contour, and boundary S is a sensor range pattern. An optimal sensor allocation scheme relates sensor S to territory T by minimizing the vulnerability measure of S on T, subject to constaints imposed by the wasted-resource measure.

1.2 QUADRATIC OPTIMIZATION OF THE VULNERABILITY FUNCTION

The process of minimizing a vulnerability function belongs to the generic class of problems known as quadratic optimization. The process is quadratic because it involves the calculation of areal integration, rather than the simplex formed by a linear system.

1.2.1 The Vulnerability Objective Function and the Constraint Sets

Given a set $Z = \{\rho_1, \ldots \rho_m\}$ of sensor envelope patterns and a set $T = \{e_1(\beta), \ldots, e_n(\beta)\}$ of emplacement-constrained territorial boundaries, minimize the vulnerability expression:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \int e_i(\beta) - \int \xi_i(\beta)\rho_j \qquad (1.4)$$

subject to the set of constraints:

$$\int \rho_j - \int \xi_i(\beta)\rho_j < \delta_{ij} \qquad (1.5)$$

$$d(\, pos(\rho_i), pos(\rho_j)\,) \; > \; \alpha_{ij}\,[\, rad(\rho_i) \; + \; rad(\rho_j)\,] \qquad (1.6)$$

Constraint set (1.5) is designed to control sensor overkill, whereas (1.6) is provided to control redundancy (sensor overlap). The constraints are logically compatible, because

by shrinking the overkill area, the overlap area is correspondingly reduced. The problem lies in balancing the constraints with the vulnerability function (1.4), since decreasing the vulnerable area frequently entails a corresponding increase in both overkill and overlap.

1.2.2 Optimal Sensor Allocation

Definition 1.2.2 An *optimal sensor allocation* is defined to be a mapping from a set of sensors to a set of territories, which minimizes expression (1.4), while conforming to constraints (1.5-1.6).

It is interesting to note that breadth-first search may be used to minimize expression (1.4), without resorting to the constraint sets. Such a brute force approach, although guaranteed to find the optimal solution, is combinatorially prohibitive and does not avail itself of knowledge to control the search space. This is where the constraint sets (and perhaps other unspecified knowledge sources) may facilitate search.

1.2.3 The Motivation for Fast Intersection and Integration Algorithms

It is evident from expressions (1.1) and (1.4) that fast algorithms are required to: a) intersect boundaries; and b) compute the area contained by boundaries. The minimization of expression (1.4) subject to constraint inequalities (1.5-1.6) is a research topic currently under investigation (Buchberger's method of Groebner bases may provide leverage), but it is clear that the optimization process requires polynomial-time mathematical techniques to intersect and integrate digital boundary data. The remainder of the paper is concerned with the development of algorithms to perform these utility functions.

2. THE BOUNDARY INTERSECTION ALGORITHM

2.1 ASSURING COUNTERCLOCKWISE ORIENTATION: ALGORITHM CCW

The boundary intersection algorithm relies upon the computer having a sense of "left" and "right" handedness as it traverses a simple digital contour. The handedness sense is imparted to the computer by assuring that the contour is oriented in a counterclockwise (ccw) direction, because such an orientation insures the "inside" of the contour is always to the left, and the "outside" always to the right.

Algorithm CCW guarantees that the digital boundary of a closed contour is in fact oriented in a counterclockwise fashion from the head of the boundary to the tail. The algorithm computes a coordinate p (in the boundary linked list) with maximum abscissa, along with its predecessor pred(p) and successor succ(p) coordinates (Fig. 2). The difference in the ordinates of p and pred(p) is computed, as is the difference in the ordinates of succ(p) and p. If either quantity is less than zero, then the boundary is oriented clockwise, and its linked list must be reversed. Otherwise, the boundary is correctly oriented in a counterclockwise direction, and no action need be taken. A formalization follows.

2.1.1 *Definition: Clockwise vs. Counterclockwise Orientation.* Given closed, simple boundary β, and coordinate $q = (x_q, y_q) \in \beta$, $\ni x_q > x_r \; \forall \; r \in \beta$. Also given the predecessor $p = (x_p, y_p)$ of q, and the successor $s = (x_s, y_s)$ of q, in boundary β. Let $\Delta y_{qp} = x_q - x_p$ and $\Delta y_{sq} = y_s - y_q$. If either $\Delta y_{qp} < 0$ or $\Delta y_{sq} < 0$ then β is oriented in a clockwise direction. Otherwise, β is counterclockwise.

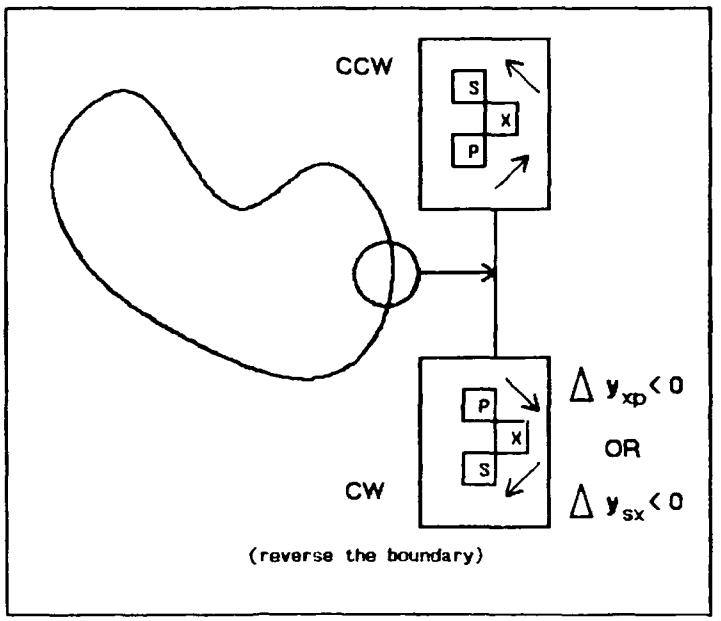


Figure 2. Algorithm CCW. A coordinate X with maximum abscissa is computed along with its predecessor P and successor S. If the difference between the ordinate of X and the ordinate of P is less than zero (or similarly for that of S and X) the boundary is ordered clockwise, and must be reversed.

2.1.2 The Formal Design Specification for Algorithm CCW

Given closed digital boundary β of length N;

BEGIN

derive $q = (x_q, y_q)$ = any coordinate in β with maximum abscissa;

$p := (x_p, y_p)$ = the predecessor of q in β;

$s := (x_s, y_s)$ = the successor of q in β;

$\Delta y_{qp} := y_q - y_p$;

$\Delta y_{sq} := y_s - y_q$;

IF $\Delta y_{qp} < 0$ OR $\Delta y_{sq} < 0$ THEN $\beta :=$ (reverse β);

END

2.1.3 The Complexity of Algorithm CCW

Algorithm CCW is a single-shot process which requires one pass over a boundary of length N coordinates to locate a coordinate with maximum abscissa. Since a destructive smash function may be used to reverse the order of a boundary which is ordered clockwise, no additional memory requirements exist, nor is query time complexity applicable.

Preprocessing: O(N)
Space: None required
Query Time: Single-shot algorithm; no queries required.

2.2 CROSSING DETECTION BY THE METHOD OF HANDEDNESS SWITCHING

In the last section we saw that algorithm CCW orients a boundary in a counterclockwise direction. The significance of the algorithm is that "left" and "right" handedness are discernible as a contour boundary is traversed from head to tail. Intuitively, a crossing of two contours occurs when information is detected on the left (right), and either simultaneously or subsequently detected on the right (left). The next crossing of necessity must now switch back to the left (right) again. Handedness switching is repeated every time a new crossing occurs during a counterclockwise traversal of the boundary, until the tail is encountered (Fig. 3). The technique is proficient at ignoring degenerate tangent behavior, because handedness does not switch at the intersection of tangential boundaries.

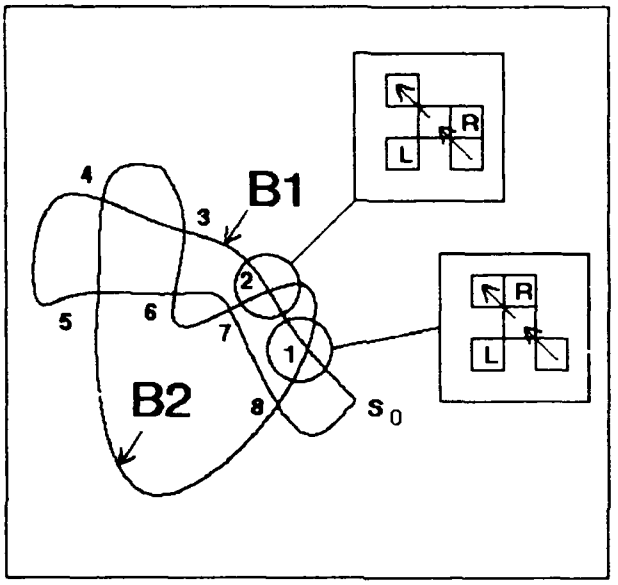


Figure 3. Handedness Switching. Boundary B1 is traversed in a counterclockwise direction, and crossings with boundary B2 are sought. At location 1, information external to B1 is detected on the left (inside), followed by more on the right. We therefore know a crossing has occurred. Furthermore, we anticipate that the next crossing will be ordered from right to left, which is in fact the case at location 2.

2.2.1 Embedding Two Digital Boundaries Upon a Bitmap

A *bitmap* M_b is a rectangular array of coordinates in the XY-plane, where each coordinate (x,y) of M_b has a binary value, denoted v(x,y), which is either 0 or 1.

A bitmap is *cleared* if and only if $v(x,y) = 0 \ \forall (x,y) \in M_b$.

A boundary β is *embedded* upon bitmap M_b if and only if $v(x,y) = 1 \ \forall (x,y) \in \beta$.

Given boundaries $\beta 1$ and $\beta 2$, define the *union embedding bitmap* $\beta 1\beta 2$.bm to be a bitmap such that:

a) $(x,y) \in \beta 1 \Rightarrow (x,y) \in \beta 1\beta 2$.bm
b) $(x,y) \in \beta 2 \Rightarrow (x,y) \in \beta 1\beta 2$.bm

c) $v(x,y) = 1 \; \forall (x,y) \in \beta 1, \beta 2$; $v(x,y) = 0$ otherwise.

Definition 2.2.1 The *handedness switching crossing set* of boundary β_1 with respect to boundary β_2, denoted C_{12}, is the set of crossings encountered when traversing boundary β_1 in the context of union embedding bitmap $\beta_1\beta_2$.bm. The set C_{12} is sequenced in the order in which the crossings are discovered during a counterclockwise traversal of β_1. The *ordinal number* of crossing c_i, denoted ord(c_i), is the sequence number of c_i in C_{12}.

2.2.2 The Formal Design Specification for the Crossing Detection Algorithm

Given closed contour boundaries β_1 and β_2, and the union embedding bitmap $\beta_1\beta_2$.bm;

```
UNTIL (Null β1) do
        Let p : = (car β1);
        Let neighbors : = the set of non-zero 8-connected neighbors of p in β1β2.bm;
        Let t1triplet : = {pred(p), p, succ(p)};
        Let t2elements : = neighbors - t1triplet;
        Let len: = (length t2elements);
        IF len > 0 AND (NULL lasthandedness)
        THEN lasthandedness : = handvector(p, t2elements);
        IF len = 1          and IF handedness(car t2elements) = opposite(lasthandedness)
                                    crossings : = (cons p crossings)
                                    handedness : = opposite(lasthandedness);
                                ELSE assert "p is tangent to β2";
        ELSE IF len>1       and IF for some point p' in t2elements
                                    handedness(p') : = opposite(lasthandedness)
                                    crossings : = (cons p crossings)
                                    handedness : = opposite(lasthandedness);
                                ELSE assert "p is tangent to β2";
        β1 : = (cdr β1);
END UNTIL;
RETURN crossings.
```

2.2.3 The Complexity of the Handedness Switching Crossing Detection Algorithm for a Closed Digital Contour Containing N Boundary Points

Preprocessing: None
Space: O(N), to store the boundary
Query Time: O(N), one-time boundary traversal

2.3 THE CROSSING ISOMORPHISM TABLE

2.3.1 The Crossing Correspondence Theorem

Theorem 2.3.1 A crossing detected by handedness switching when traversing contour β_1 in embedding bitmap $\beta_1\beta_2$.bm corresponds to one detected when traversing contour β_2.

Proof: Let q be the crossing detected when traversing β_1. Without loss of generality, assume that some element p of β_2 was to the left of q. But this means that some previous element p' of β_2 was to the right of some previous element q' of β_1, by the definition of handedness switching crossing detection. But if p is to the left of q, and p' is to the right of q', then q is to the right of p, and q' is to the left of p'. This means that either p or p' is a crossing in β_2.

2.3.2 The Crossing Isomorphism Theorem

It should come as no surprise that the crossing set encountered by handedness switching when traversing one contour is the same as that encountered when traversing its partner, within the context of the union embedding bitmap. This section provides a formal proof of this concept, but first a definition is in order.

Definition 2.3.2. Let β_1 and β_2 be digital boundaries embedded in bitmap $\beta_1\beta_2$.bm. Let C_{12} be the set of crossings detected by handedness switching when traversing C_1 with respect to C_2, and C_{21} be the set of crossings detected when traversing C_2 with respect to C_1. A *crossing isomorphism table* $\emptyset(C_{12}:C_{21})$ is a mapping from C_{12} to C_{21} which satisfies the following:

$$\emptyset(c_i) = c_j \text{ iff } d(c_i,c_k) > d(c_i, c_j)\ \forall\ j \neq k\ ;\ c_i \in C_1;\ c_j, c_k \in C_2.$$

Theorem 2.3.2 The set of handedness switching crossings encountered when traversing one curve with respect to another contour is the same set encountered when the roles of the contours are reversed.

Proof. Let C_{12} be the crossing set of contour C_1 with respect to contour C_2, and C_{21} be the crossing set of contour C_2 with respect to C_1. Let $\emptyset(C_{12}:C_{21})$ be the crossing isomorphism table defined on C_{12} and C_{21}. We will show that $\emptyset$ is both onto and one-to-one.

a) Proof that C_{12} is onto C_{21}. Assume there exists an element p' of crossing set C_{21} which does not correspond to some element of crossing set C_{12}. Recall that p' is derived by detecting elements of C1 which possess opposite handedness when traversing C2 in a ccw direction. But this means that the same disparity in handedness must have been encountered when traversing C_1, from theorem 2.3.1. Hence some crossing in C_{12} must correspond to p'. Therefore, by *reducio ad absurdum*, C_{12} is onto C_{21}.

b) Proof that the correspondence $\emptyset$ between C_{12} and C_{21} is one-to-one. Assume distinct crossings p and q $\in C_{12}$, $\ni \emptyset(p) = \emptyset(q) = r$ for some $r \in C_{21}$. Without loss of generality, assume that crossing r was discovered as a result of encountering a single element of C_1 on the left of r. But this implies that both p and q are to the left of r and both belong to C_1. However, since only a single element of C_1 was to the left of r, then p must be the same crossing as q, which contradicts our original assumption. Therefore $\emptyset$ is one-to-one.

The crossing isomorphism table is crucial to the intersection process. The fact that two boundaries are oriented in a counterclockwise fashion ensures that the intersection can always be found by traversing to the left when encountering the tail of an arc in one

boundary, because it corresponds to the head of an arc in the other boundary. This technique is proficient even for intersections which consist of multiple pieces.

Since the amount of memory consumed by the crossing isomorphism table is generally negligible, it is suggested that the table be computed and stored during a preprocessing step, for those pairs of boundaries which are expected to remain static. This obviates a query time traversal of both boundaries to discover crossings and to establish the correspondence by ordinal number.

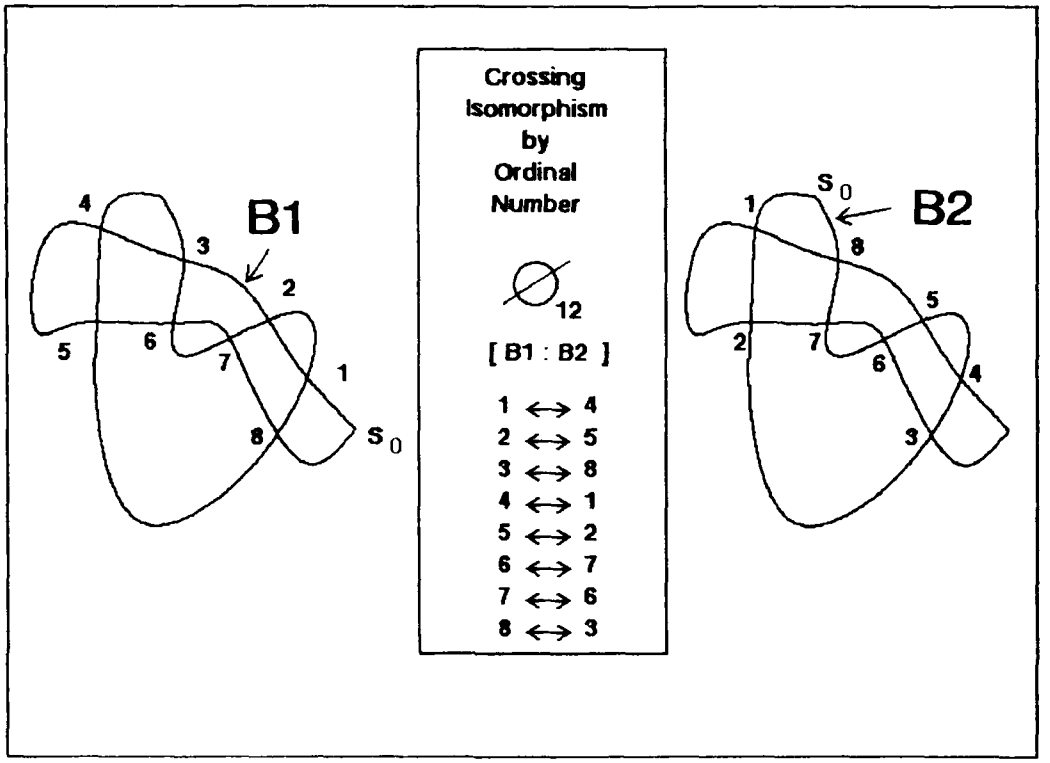


Figure 4. An example of a crossing isomorphism table (ordinal numbers correlated). While traversing boundary B1 in a counterclockwise direction, eight crossings are discovered by the method of handedness switching. A boundary element is assigned an ordinal crossing number when a switch in handedness occurs. The same process is repeated for boundary B2. The two sets of crossings are correlated by a distance metric to produce the table, which is then used as an index to collect alternating arcs from B1 and B2. Referring to the diagram on the left, closed contour 1-2-7-8-1 is produced first; subsequently closed contour 3-4-5-6-3 is discovered.

2.4 INTERSECTION: ALTERNATING ARC COLLECTION

2.4.1 Alternating Arc Collection Generates the Intersection

Note that the intersection of two closed contours consists of the union of arcs which lie upon either boundary and are wholly contained within the confines of the other boundary. The head of each arc is an entry into the interior of one of the boundaries, and the tail is an exit. Furthermore, the tail of an arc exiting the interior of one boundary is in fact the head of another arc entering the interior of the other. Recall that the endpoints of

the arcs are crossings detected by handedness switching, and that the crossing isomorphism table is available to reflect the ordinal number cross-referencing correspondence between contours. Therefore, the arcs can be collected systematically in an alternating fashion to form distinct planar graphs which are themselves closed contours, and which *collectively form the intersection of the two contours* (Fig. 4).

2.4.2 The Formal Design Specification for Intersection by Alternating Arc Collection

Given ccw-oriented digital closed contours β_1 and β_2, and the crossing isomorphism table sets C_{12} and C_{21}, where Cij is the car-sorted set of 4-tuples {(*ordnum*-β_i; *ordnum*-β_j; *crossing-point*-β_i; *crossing-point*-β_j)};

Arbitrarily select β_1 to traverse in a ccw direction;

```
        cnt : = 1; len1 : = length(C12);
        test : = insidep(head(β1), β2]
        IF test = false THEN pointer : = car (C12) ELSE pointer : = car (last (C12)) ;
        boundary : = (car pointer)
LOOP    cnt : = cnt + 1
        cnt : = cnt MODULO 2
        (cond ( (zerop cnt)
                        (setq refset C12)
                        (setq altrefset C21))
              (t        (setq refset C21)
                        (setq altrefset C12)))
        (setq point1 (assoc (car pointer) refset))
        (setq newcnt (add1 (car pointer)))
        (cond ( (igreaterp newcnt len1)
                        (setq newcnt 1)))
        (setq point2 (assoc newcnt refset))
        (setq portion (nconc1 portion (list (add1 cnt) (car point1) (car point2))))
        (setq pointer (assoc (cadr point2) altrefset))
        (cond ( (and (eq cnt 1) (eq (car pointer) boundary))
                        (setq alt-arc-list (nconc1 alt-arc-list portion))
                        (setq portion nil)
                        (setq boundary (newanchor boundary len1 prize))
        (cond ( boundary
                         (setq cnt 1)
                         (setq pointer (assoc boundary refset)))
              (t         (return alt-arc-list)))))
        (go LOOP))
```

2.4.3 The Complexity of Intersecting Two Closed Contours of Lengths N1 and N2, by the Method of Alternating Arc Collection, Using the Crossing Isomorphism Table $\emptyset(C_{12}:C_{21})$

The preprocessing required for intersection by the method of alternating arc collection involves one pass each over boundaries β_1 and β_2, respectively of lengths N1 and N2 coordinates, to produce isomorphism table $\emptyset(C_{12}:C_{21})$. The arcs encompassing the intersection are extracted from the boundaries in time proportional to the size of the isomorphism table.

Preprocessing: $O(N1 + N2)$, to generate $\emptyset(C_{12}:C_{21})$
Space: $O(N_1 + N_2 + \text{len}(\emptyset(C_{12}:C_{21})))$, to store the arcs specified by $\emptyset$
Query Time: $O[\text{len}(\emptyset(C_{12}:C_{21}))]$

3. INTEGRATION BY VERTICAL RAY PIERCING

3.3.1 The Morphology of a Closed Curve Affects the Complexity of Computing Its Integral

The vertical ray piercing integration (VRPI) algorithm computes the area bounded by a planar graph. The graph may be non-convex with an arbitrary number of exterior concavities, may be multiply-connected, and may be comprised of multiple pieces. The algorithm logic migrates by ascending abscissa along a sorted boundary list, and pierces the ordinate points (also in ascending order) with a vertical ray drawn upwards from the x-axis. The odd-numbered encounters are treated as entry points into the graph, whereas the even-numbered encounters are exits. The lengths of the entry-exit pairs are computed for each abscissa, and a running sum is accumulated until the maximal abscissa is processed (Fig. 5). The final sum is returned as the integral of the planar graph.

Definition 3.3.1 Let Y_n be the sorted (in ascending order) set of ordinates lying over x_i, where x_i is the abscissa of some coordinate in boundary β. β is said to possess a *concavity* above x_i iff $\exists\, y_j \in Y_n \ni y_j - y_{j-1} > 1$ and $y_{j+1} - y_j > 1$.

The number of boundary concavities affects the complexity of the ordinate entry-exit processing. Let X denote the set of unique abscissas occurring in elements of boundary β. A concavity may exist due to non-convex structure of the boundary itself, or because the boundary is multiply-connected. A boundary with no concavities has at most one entry-exit ordinate pair for each distinct abscissa in X; one with one concavity has two entry-exit pairs for each abscissa lying beneath the concavity; in general, a boundary with m concavities requires m + 1 ordinate entry-exit computations for each distinct abscissa lying below the m concavities. This concept is formalized with the following theorem:

Theorem 3.3.1 Let boundary β contain m concavities. Let X(0) denote the set of unique abscissas which lie beneath no concavities, X(1) beneath one concavity, ..., X(m) beneath m concavities. Then the total number ω of entry-exit ordinate computations performed during integration by vertical ray piercing is equal to:

$$\omega = \sum_{i=0}^{m} X(i) * (i + 1) \qquad (1.7)$$

Proof. Proceed with a counting argument. Each element of X(0) requires exactly one ordinate entry-exit computation, because the boundary does not fold over itself above any element of X(0). Each element of X(1) requires exactly two computations, because the boundary folds over itself once above each element of X(1). In general, an element belonging to X(m) requires exactly m + 1 ordinate entry-exit computations. Across the set of all unique abscissas, the total number of computations is expressed by equation (1.7).

3.3.2 The Formal Design Specification for VRPI

Sort the boundary, using abscissa as the primary key, and ordinate as the secondary key. Let XMIN be the minimum abscissa, and XMAX the maximum abscissa.

```
integral : = 0;
        For xi = XMIN to XMAX
        BEGIN
                YSET : = {yj, j = 1,k}              (* The set of ordinates above xi *)
                UNTIL (NULL (cdr YSET))
                        Let ylast : = (car YSET); ynew : = (cadr YSET);
                        orddiff : = ynew - ylast
                                IF orddiff = 1 then integral : = integral + 1
                                ELSE integral : = integral + orddiff + 1;
                        YSET . = (cddr YSET);
                END UNTIL;
                        IF (NULL YSET) return integral
                        ELSE
                        Let ylast : = ynew; ynew : = (car YSET); orddiff : = ynew - ylast;
                                IF orddiff = 1 then integral : = integral + 1
                                ELSE integral : = integral + orddiff + 1;
        END BEGIN.
RETURN integral.
```

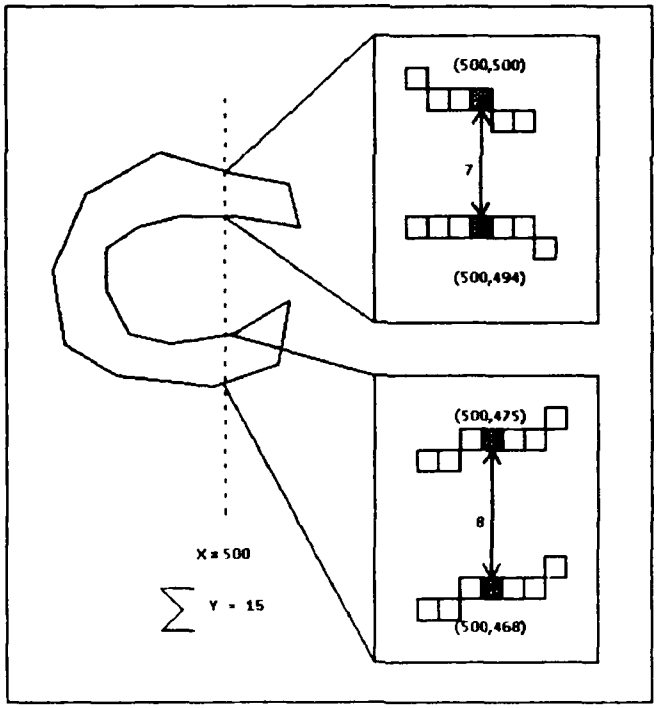


Figure 5. Vertical ray piercing for one abscissa, within the area of a single concavity. All such results are summed from minimal to maximal abscissa to derive the integral of the closed region.

3.3.3 The Complexity of Integration by Vertical Ray Piercing

Let boundary β be of length N. The computational complexity of VRPI is:

Preprocessing: O(N*logN), to sort the boundary; primary key x; secondary key y
Space: O(N)
Query Time: O(N)

Note that on a parallel architecture, if a processor is dedicated to each unique abscissa, the time complexity of the vertical ray piercing algorithm is of order m + 1, where m is the maximum number of concavities lying above any single abscissa. That is say that m + 1 entry-exit differences must be computed for each element of X(m).

3.3.4 A Comparison of the Complexity of VRPI with Other Areal Integration Techniques

There are other techniques to compute the area contained by closed boundaries; the field of numerical analysis abounds with integration methods. However, these techniques generally require additional processing to treat the topological phenomena which result when intersecting digital boundaries of general complexity. The techniques may also require calculations for every point on the boundary, rather than at each unique abscissa. These calculations may entail multiplication, which is more compute-intense than addition. Vertical ray piercing integration treats these issues as follows:

VRPI accomodates intersections which lie in *pieces*, with no additional processing. Other algorithms may expect that each piece has a separate, gap-free boundary, with a handle into the head of the linked list representing the boundary.

VRPI accomodates *multiply-connected sets*, with no additional processing. Other algorithms may require that each hole be treated as a separate object (for which a predefined handle must exist). The area of each hole must be computed and subtracted from the area of the outer boundary.

VRPI requires a *preprocessing sort* of the boundary, which other integration algorithms may not require. However, the sort buys superior query time performance for two reasons. First of all, the number of VRPI calculations is proportional to the number of *unique* abscissas, and not the total number of abscissas. Second, *arithmetic differences* are computed above each *unique* abscissa, which is far less costly than performing multiplications at *every* boundary element.

4. SUMMARY

The sensor-to-territory allocation problem has been posed in terms of minimizing a quadratic vulnerability function subject to a set of quadratic constraints. It is shown that boundary intersection and integration are vital subproblems: the intersection may pathologically lie in pieces, be multiply-connected, and is not guaranteed to be contiguous Because the sensor-to-territory mapping is potentially prohibitive from a combinatoric standpoint, it is important at least to have fast algorithms at hand to perform intersection and integration. New linear query time algorithms are presented for each operation. The intersection algorithm relies upon two new techniques: one which guarantees counterclockwise orientation of a closed boundary; and one which detects crossings by relying upon a sense of handedness switching during boundary traversal. The integration

algorithm utilizes a technique called vertical ray piercing, which initiates a migration by ascending abscissa through a sorted boundary list. The sorted set of ordinates over a unique abscissa is pierced from below, using parity logic to accumulate the integral. Future directions of the research include addressing the problem of efficiently solving a set of second order equations which serve as constraints during minimization of the vulnerability function.

ACKNOWLEDGMENTS

Insights into the theory of generalized digital contour intersection were made possible as a result of discussions with Chris Bogart, Rich Antony, and Cathy Lamanna. A true appreciation of the problem was not forthcoming until the author was provided by Mr. Bogart with a Byzantine set of real-world digital contour data. Thanks to Gordon Novak for suggesting a comparison of the complexity of the vertical ray piercing algorithm with that of other areal integration techniques.

REFERENCES

Buchberger, B., G.E. Collins, and B. Kutzler, 1988: "Algebraic Methods for Geometric Reasoning", Annual Review of Computer Science, Volume 3, Annual Reviews Inc., 95-104.

Edelsbrunner, H., 1987: Algorithms in Combinatorial Geometry, Springer-Verlag, New York.

Preparata, F. P., and M. I. Shamos, 1985: Computational Geometry, An Introduction, Springer-Verlag, New York.

Rosenfeld, A., and A. C. Kak, 1982: Digital Picture Processing, Second Edition, Volume 2, Academic Press, Orlando FL.

Sedgewick, R., 1983: Algorithms, Addison-Wesley, Reading MA.

Representation Issues in the Design of a Spatial Database Management System

Richard Antony
U.S. Army CECOM Center for Signals Warfare
Vint Hill Farms Station
Warrenton, Va 22186-5100

Abstract

Representation issues and evaluation criteria are presented regarding the design of a general purpose database management system to support complex spatial reasoning. For problem domains that require extremely large amounts of data, the database design must simultaneously consider 1) storage efficiency, 2) search efficiency and 3) problem solving efficiency. Six critical representation issues are discussed that impact these three aspects of database efficiency, classes of representations for point, line and region features that satisfy the various representation issues are discussed. A hybrid representation approach is recommended that balances the strengths and weaknesses of several representations to achieve the above efficiency goals. Finally, the framework for an efficient spatial DBMS (SDBMS) is proposed that is based on these general principles.

1. Introduction

Numerous complex computational domains, including battlefield data fusion, image understanding, assisted target recognition and autonomous vehicle control, demand an underlying element of spatial reasoning. Battlefield data fusion, for instance, requires the correlation of sensor measurables against both the current situation assessment and potentially massive amounts of domain data, such as 1) object-oriented knowledge (e.g., tables of equipment, tables of organization), 2) spatially-oriented natural, cultural and abstracted features (e.g., rivers, roads, line-of-sight, respectively) and 3) dynamic target location/attribute data.

Spatial databases are often developed in a bottom-up manner based on a single underlying representation for points, lines and regions. With relatively small databases and fast processors, the combinatorics generated by potentially inefficient representation, search and evaluation operations may be of little concern. However, for very large databases, control of the search space size and computational requirements to support real time, complex automated reasoning is of considerable importance. This paper discusses a top-down database design approach that simultaneously addresses 1) storage efficiency, 2) query efficiency and 3) problem solving efficiency.

Just as an indexed relational database file facilitates access to data that is organized along natural search dimensions (e.g., alphabetically or numerically indexed keys), a true two-dimensional representation offers inherently more efficient access to spatially organized knowledge than a non-spatially organized representation. Similarly, a tree-structured search for a specific object among a collection of hierachically-organized objects is more efficient than a linear search though an entire object database.

Storage efficiency is achieved through some form of data compaction. *Search efficiency* requires effective control of the size of the search space for the range of typical database queries. *Problem solving efficiency* can be achieved by minimizing both the overall search space size and the required data transformation and manipulation that may, in part, be caused by the data compaction process.

A natural tradeoff tends to exist between memory efficiency and query / problem solving efficiency; for spatial data, compact representations often do not support efficient search and abstraction. For example, the chain code for the boundary of a region, although memory-efficient, is a computationally expensive form for performing set operations. Thus, an ideal representation should be relatively compact and yet not require excessively large search spaces or complex, computationally-intensive manipulations to develop typical products.

Because of its broad processing requirements, battlefield data fusion will be used throughout this paper to illustrate many of the database issues. Battlefield data fusion is the process of creating and maintaining a coherent "picture" of a dynamic situation based on limited domain observables from one or more sources. A fundamental database query that supports asynchronous data fusion takes the form:

Find database element(s) that is (are) *near* $(f_1, ..., f_N, g(x,y), t)$

where f_n are features or attributes, g represents a location uncertainty function and t represents a time tag. Observation attributes include frequency, velocity, cross section, altitude, modulation characteristics, and so forth; *near* is interpreted within the context of an arbitrary metric.

The spatial and temporal aspects of observations are key elements of the fusion process. *Near* in terms of (x,y,t) may require intersection of the function g( ) with relatively static (time-invariant) database elements, such as terrain, elevation, soil type, trafficability, cultural features, waterways, or relatively dynamic (time-sensitive) database elements such as individual target tracks, groups of targets or Named Areas of Interest (NAI). The concept of *near* can be highly situation and context-sensitive and may depend on some state of the world, observation class, target velocity, report rate, resolution, and the specific metric (e.g., Euclidean distance).

The underlying static domain database provides the knowledge that constrains activities and interpretations, such as target class-dependent limitations on trafficability, observability, communications and access. The non-static data provides the historical and time-varying domain knowledge (e.g., weather, target track files) that forms the basis for interpretation of new observations and refinement of previous observations.

No automated system developed to date has achieved the spatial reasoning capability of a skilled human analyst. The ease with which military analysts perform spatial reasoning, in

large measure, is the result of very natural, fully registered, true two-dimensional spatial representations provided by maps and acetate overlays. The manual *grease pencil and acetate overlay method* used by human analysts is, in fact, a conceptually powerful spatial problem solving paradigm. The representation medium provides a uniform, true two-dimensional spatial representation that allows effective spatial focus-of-attention, highly intuitive manipulation of features, efficient development of Euclidean metrics and set operations performed virtually "by inspection" (e.g., human perception facilitates the interpretation of closed lines as representing the enclosed two-dimensional area).

Acetate overlays provide natural, context-conditioned conceptual planes which afford additional search space reduction. For instance, when reasoning about primary road networks, acetate overlays containing other feature classes need not be searched. Newly abstracted spatial features (e.g., the result of set operations among existing features) are stored in the same uniform, fully registered representation.

Many current data fusion systems and other semi-automated systems rely on conventional relational database management systems (DBMS). Future highly automated reasoning systems will require extremely large databases that support efficient spatial query and spatial problem solving at multiple levels of abstraction.

The advantages of emulation of both the problem solving style and the database form of the acetate and overlay approach for automated spatial reasoning is often overlooked, in part because the manual process is viewed as a rather crude, low resolution, low accuracy technique. However, the DBMS analog of acetate overlays need be neither low resolution, nor low accuracy; a database form motivated by the human spatial reasoning metaphor is outlined in Section 3.

2. Database Issues / Requirements

In general, database query requirements include 1) location of a specific data item, 2) range query / windowing, 3) set operations among multiple data sets and 4) evaluation of data relationships (parametric, Euclidean and non-Euclidean distance, generalized metrics). These tests reduce to search and test for equality, non-equality and containment. The test constraints or conditions effectively represent a template through which data is viewed. Thus, in general, database query can be viewed as a templating operation.

The conditions or templates can be as simple as "is point (x,y) in the database" or as complex as "find all forested regions with elevations above 1000m that are within 100km of a major population center west of the Mississippi river" or "find all airports with 3 or more runways east of the Continental Divide that 1) are not surrounded by industrial or residential development, 2) are at elevations greater than 500 m[illegible]d 3) support 3 to 5 major air carriers" The second example has a strong geogr[illegible]ormation system (GIS) flavor, while the latter has a mixed object-oriented and spatial-oriented character. True two-dimensional spatial representations support efficient spatial oriented search; true object-oriented representations support efficient search and manipulation of complex object relationships

In a highly dynamic environment, in order to support a broad variety of applications, a database must maintain both context-sensitive and context-insensitive data Context-sensitive data includes the current situation representation and previously

developed abstractions, while context-insensitive knowledge includes much of the relatively static factual and procedural reasoning knowledge base.

For instance, basic elevation data is context-insensitive knowledge that is used in the generation of numerous context-sensitive products. For instance, line-of-sight and path-planning algorithms use elevation data to produce highly context-sensitive products which depend on 1) observation position and range limit, and 2) weather, soil type and unit class, respectively. Context-sensitive products such as river crossing sites within a specified area of interest should ideally be evaluated on-line based on unit class, recent rainfall, general weather conditions, current river depth and flow rate, vehicle-dependent trafficability on the river banks, potential bridge erection time, logistics considerations, mission priority and so forth.

As opposed to prestoring all possible context-sensitive databases (which may not even be enumerable), the context-insensitive database can be abstracted to generate specific products (based on current context) as required. Such a capability enhances both the storage efficiency and robustness of the database. Thus, storage efficiency, search efficiency and problem solving efficiency become key factors in the selection of database organization.

2.1 Representative Classes of Spatial Reasoning

In this section a number of representative automated spatial reasoning tasks are discussed that require database support. A simple template-oriented database query has the form:

Find (n / all) (objects, / features) **set function** (objects, / features) **conditioned on** (objects, / features)

Examples of this form include:

Find one truck on road A (intersect) within spatial region B (intersect).

Find all secondary roads (union) or Trafficability Index 3 regions (intersect) east of Rockville.

In general, set operations can be concatenated to yield more complex forms.

As an application of intersection, consider the refinement of a target's location by correlation of sensor derived information with context-conditioned domain knowledge. For instance, a particular target class may be known to prefer operating within 50 m of a secondary road that is within a lightly forested area which provides moderate concealment. Figure 1 shows the development of the refined target location based on the successive intersection of two independently measured error ellipses, with regions that are within 50 m of a secondary road and within lightly forested regions. The final product (Fig. 1g) provides potentially better target localization than is afforded by the intersection of only the sensor derived error ellipses (Fig. 1c). Template conditions can be applied separately or in combination using the same basic implementation mechanics.

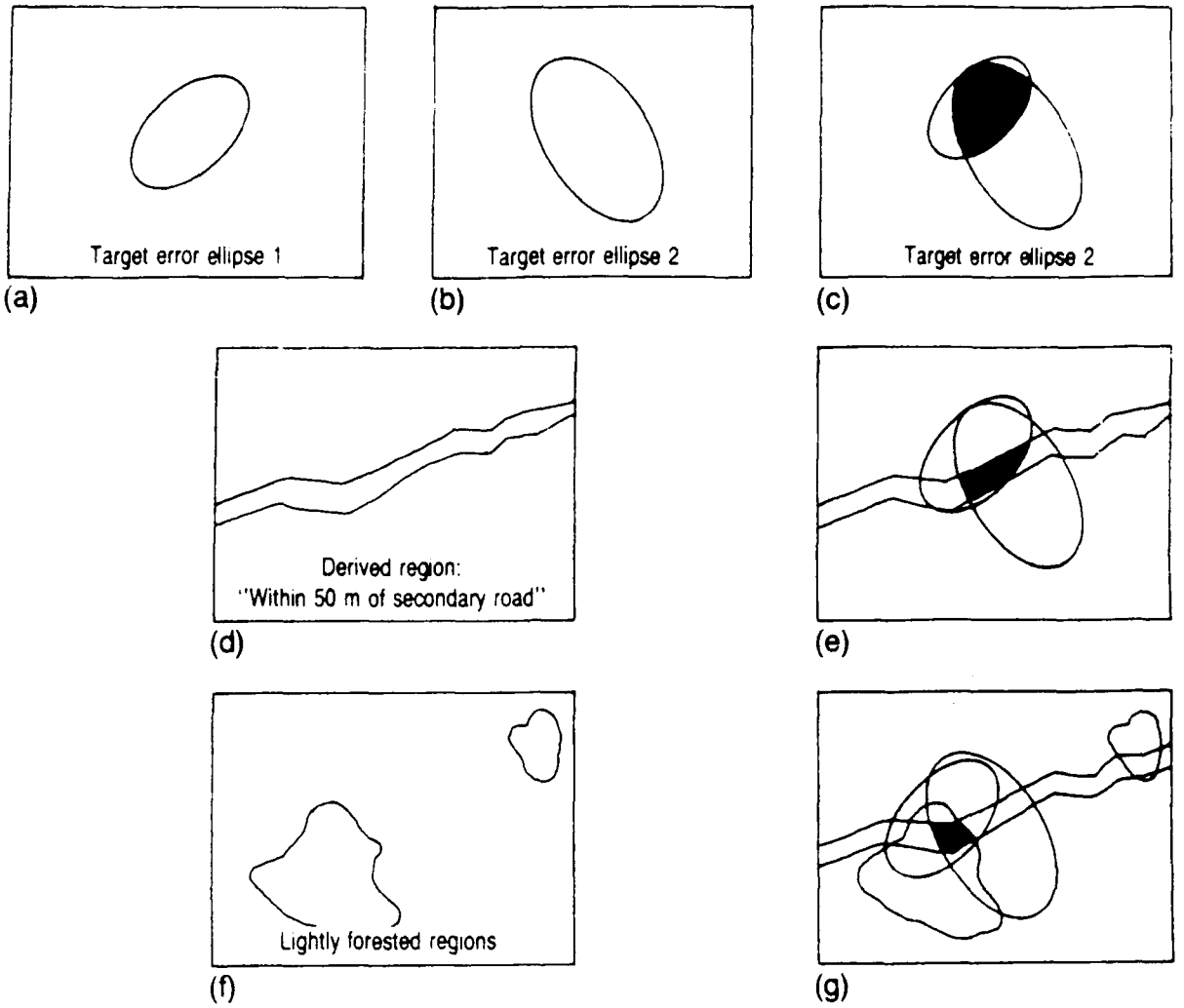


Figure 1 Initial target location estimate (a) is refined by intersection with an independently measured uncertainty function (b) as shown in (c); the resulting region is intersected with the "region-grown" road network (d) as shown in (e) and the cummulative intersection with the lightly forested region (f) is shown in (g). The shaded region is the predicted target location following each corrrelation operation.

The union of points, lines and regions yields the composite of all addends and the difference yields the nodes of the minuend that are not contained within the intersection. Containment can be easily implemented by observing that $A \subset B$ if $A \cap B = A$.

In general, fuzzy modifiers and qualifiers (adverbs, adjectives and associated phrases) modify generalized metrics or other modifiers. Metrics of interest to spatial reasoning include generalized distance (e.g., Mahalanobis distance, correlation ratio between data and templates, shape, size and speed). While Boolean functions provide a natural representation for absolute modifiers or conditionals (e.g., *nearest, before, more, to the left of, higher, closer*), fuzzy membership functions offer a natural representation for relative modifiers (*near, very, almost, small, fast*).

Most absolute modifiers can be implemented as a simple Boolean function. Relative modifiers, on the other hand, tend to be context-sensitive, requiring a family of conditional fuzzy membership functions. Fuzzy modifiers (and their context-sensitive membership functions) may depend on the state of a process, the distance from a goal state, the current weather conditions, and so forth.

The use of generalized conditional membership functions effectively allows the database to support general probabilistic, fuzzy and multi-valued logic set operations. Using a simple region-growing approach, database features of various membership values can be developed; these generalized fuzzy regions can then be manipulated by the standard Boolean set operators.

For instance, consider the query:

Find all vehicles *near* Region A

In this case, *near* is the fuzzy distance metric described by a conditional membership function. The fuzzy intersection can be computed as the Boolean intersection of the target error ellipse with the fuzzified Region A ; analogously, the fuzzy intersection could be developed as the Boolean intersection of Region A with the fuzzified target error ellipse. Because of target location uncertainty, map registration errors and Region A representation errors, fuzzy set operations are a critical requirement in many real world problem domains [Antony, 1989].

Spatial templating involves the recognition of spatial patterns of terrain, cultural features, vehicles or units. Because of the large number of rigid templates that may be required to account for variation in size, shape, spacing, orientation, contrast, observation angle and so forth, the use of rigid templates tends to require large storage and excessive computational resources. The use of *fuzzy semantic templates*, on the other hand, can dramatically reduce the total number of required templates.

Consider a template for a specific play in football that expresses the semantic relationship among players on the field. As an example, an "end" for a certain play may normally be "to the right and slightly forward of the quarterback". "To the right and forward" are absolute (Boolean) modifiers. Given the location of either of the entities, a spatially organized database can be readily searched for the other element. *Forward* is a dynamically evaluated absolute modifier whose meaning is determined based on the current position of the quarterback and the goal direction. *Slightly* is a context-sensitive relative distance metric that can be represented as a conditional fuzzy function. A complex template can be efficiently implemented as a set of procedural or scripted, absolute and relative database queries.Thus, relative to the quarterback, a fuzzified search window can be constructed based on the relative semantic spatial relationships among the entities and not on a strict mathematical correlation between the database and an exhaustive set of rigid templates.

As a second example of spatial templating, consider the conceptually trivial task of associating N-tuples to candidate K-member groups based on proximity. With a non-spatially-organized database, as many as (N!/(K!*(N-K)!) locational comparisions need to be made. For N = 100 and K = 5, 10^7 possible tests may be required

A simple database search algorithm that avoids the combinatorial explosion can be developed if the N-tuple database possesses a true spatial representation. A spatial window large enough to enclose the nominal cluster is first overlaid on a previously non-clustered N-tuple. The window centroid is then moved to each of the N-tuples that appear within the window and the total number of non-clustered N-tuples within the window is determined. For non-overlapping clusters, when the window centroid reaches the cluster centroid, the maximum number of spatially associated N-tuples will appear within the window. Thus, the cluster centroid, as well as the cluster member count has been tentatively determined. The detailed pattern and parametrics of the constituent elements can then be evaluated in greater detail. The algorithm recurses until all unassociated N-tuples have been clustered. Because of the efficiency of two-dimensional spatial operations, a potentially combinatoric search and compute-and-test task is reduced to an efficient, top-down intersection of a generalized spatial window with the spatially-organized N-tuple database.

Path planning / replanning involves boundary-following, feature-following, line-of-sight computation and generalized path development under time-varying and context-varying constraints. A generalized multiple resolution, recursive path-finding algorithm [Antony, 1986] supports both top-down global, as well as optimal single resolution problem solving.

Metric computation involves 1) Euclidean distance measures (e.g., feature separation, boundary length, percent overlap, correlation coefficient), 2) moment computation (e.g., centroid, center of gravity, area) and 3) fuzzy metric evaluation (e.g., *near, forward, high*). Efficient evaluation of such metrics is fully supported by a true two-dimensional, fully registered spatial representation.

2.2 Spatial Feature Representations

In this section a number of popular representation forms for points, lines and regions will be briefly discussed in the context of desirable attributes of a spatial DBMS.

2.2.1 Points

The most general representation of an n-dimensional point is the real-valued *n-tuple*. A *pixel-based point* representation is a grid-based, finite resolution representation where the value of a pixel is either a continuous-valued (e.g., real number) or discrete-valued (e.g., gray-scale code, color or integer-valued weight).

The *point quadtree* is a non-regular variant of the region quadtree where each data point is assigned to a separate quadtree node. Nodes are added to the tree when search for the point terminates at a leaf node without finding the data point. The size and shape of the tree is dependent on the order of insertion. Since balancing the tree and deletion of data (which, in general, requires merging nodes) are relatively expensive operations, the point quadtree is a relatively static data structure.

A *k-d tree* (2-d for two dimensions) is a binary search tree (each node has two sons) where the branching is based on alternating between x and y coordinates. In the *adaptive k d tree*, all data is presorted and stored at leaf nodes. Unlike point quadtrees, the tree shape is independent of the order of insertion and deletion.

The *MX* (or matrix) *quadtree* is a region quadtree-based representation for a finite number of discrete data points. Each data point is associated with a single pixel in a matrix grid; thus, the pixel size defines the minimum resolution that can be represented. The decomposition is regular, order-independent and a relatively dynamic data structure.

The *PR* (P for point; R for region) *quadtree* is a form of region quadtree where a quadrant is split if a new data point belongs in an existing node. The splitting continues until the data points are no longer in the same quadrant. The PR quadtree produces a unique decomposition, independent of order of insertion. However, the representation may require great depths to represent small differences between data points. Deletion may cause node collasping.

2.2.2 Lines

Lines are extended spatial features that represent the boundary of either open or closed regions (points and open lines can be considered to be degenerate regions). In general lines can be represented using either regular or non-regular spatial decompositions. In regular decompositions, two-dimensional space is subdivided into a uniform grid structure which may or may not be hierarchical. The *boundary code* is perhaps the simplest pixel-based line representation which specifies the ordered sequence of pixels that the line passes through. The *chain code* is a form of run-length encoding where only direction changes of the boundary are represented.

A number of tree-structured representations are based on a regular decomposition, as well. In the *edge quadtree*, the leaf nodes are quadrants that possess straight line segment approximations to the original line. The *MX quadtree* is a matrix (pixel-based) representation of line segments where the line thickness is effectively the pixel width. In the *PM quadtree*, each polygon vertex is a data point in a PR quadtree.

Non-regular decompositions include generating functions (e.g., polynomials), piecewise linear approximations (e.g., end point tuples) and certain tree-structured representations (e.g., strip tree).

2.2.3 Areal Representations

A polygon is a memory-efficient piecewise linear approximation of the boundary of an arbitrarily shaped closed region. The vertices of the polygon are represented, in general, by a linked-list of (x,y) coordinates. Closely related chain, boundary and border code representations are pixel-based versions of the general polygon representation. Although boundary-only representations are perhaps the most commonly employed region representations, they do not explicitly represent the region interior.

Raster, pyramid and various tree-structured forms provide true areal representation. The raster is a pixel-based representation of the boundary plus the interior of a region. The pyramid is a closely associated multiple, but fixed resolution representation. The region quadtree is a variable resolution, fixed-reference maximal block size decomposition of the boundary plus the interior of a region.

2.2.4 Region Quadtree and the Pyramid

Because the region quadtree possesses several desirable attributes (e.g., 1) regular, 2) hierarchical and 3) true areal representation), that are exploited by the proposed SDBMS discussed in Section 3, an expanded discussion of this representation form is presented. The region quadtree is a regular, recursive decomposition of Euclidean space into equal-sized quadrants, each of those into equal-sized quadrants, down to a minimum application-dependent resolution. By expanding only those quadtree nodes that are partially within a quadrant, a minimal quadtree representation results.

A simple string notation, $X_0X_1...X_i$, provides a unique node representation and defines a unique tree traversal path from the root to that node. Each term in the string, X_i, defines a specific quadrant, X, at tree depth i. X is a four-valued variable that is defined as follows:

A	northeast quadrant
B	northwest quadrant
C	southwest quadrant
D	southeast quadrant

The node-string notation is simply a radix 4 coding scheme that exhibits familiar abelian group properties with respect to addition and subtraction. The quadtree and the associated string notation is a relative, incremental coding scheme which can be thought of as two-dimensional run-length code. Thus, for all j, $0<j<i$, $(X_0X_1...X_i) = (X_0X_1...X_{i-j}) + (X_{i-j+1}...X_i)$ where + can be interpreted as either addition or concatenation. Since each term in the string takes on only four values, each term can be represented by two bits. Thus, a 32-bit string can effectively index 4×10^9 (i.e., $2^{16} \times 2^{16}$) raster level cells.

The cardinal and non-cardinal directions can be treated as functional operators on quadtree nodes (e.g., $\text{east}(A_1B_2A_3C_4) = (A_1B_2A_3D_4)$). From symmetry, concatenated opposing operators cancel each other, while non-opposing operators yield the non-cardinal operators.

A simple nearest-neighbor move rule and efficient spatial windowing algorithms make the quadtree a powerful hierarchically-organized spatial representation [Antony, et al., 1987]. Virtually all quadtree-related algorithms require only integer arithmetic; many of the algorithms can be implemented using simple table-lookup.

Since each level of the tree (denoted by i) represents a node resolution size proportional to 2^{-i}, absolute and relative map coordinate computation for a quadtree node is straightforward (e.g., A_i represents a translation *north* by 2^{-i}dy and *east* by 2^{-i}dx, where dx and dy are the x and y dimensions of the map root node) Since the region quadtree recursively decomposes homogeneous regions into maximal quadtree blocks, the bulk of the interior of a large region is represented by relatively few large nodes; the remaining interior is represented by progressively larger numbers of smaller resolution blocks

A pyramid (which can be implemented as a complete quadtree) with its inherent multiple resolution levels offers efficient top-down directed tree search potential (e.g., $A_0C_1D_2 > A_0C_1D_2X_3 > A_0C_1D_2X_3X_4$, for all X). For instance, consider the interrogation of a map knowledge base to determine (1) if a bridge exists within a particular block of (n x n) raster level cells and (2), if so, which resolution cell contains the bridge. The first question can be answered by merely interrogating the parent node of the desired raster cell block. In a conventional single level of abstraction raster representation scheme, the cells are searched sequentially until the existence of a bridge is discovered. In the worst case, all n^2 cells must be interrogated even though that no bridge exists in the region.

The second question can also be answered efficiently in a hierarchically organized database since the search for a single bridge in the region involves only four branches that emanate from a single node at each level in the tree. Thus, while the non-directed (exhaustive) raster search requires an exponential search space size (on the order of 4^i where $n = 2^i$), the directed quadtree-based pyramid search involves only a polynomial search space size (on the order of 4i), where i is the depth of the tree representation.

Since regular, grid-based representations are highly sensitive to translation of the underlying reference system, a simple comparision of storage efficiency with other region representations (e.g., boundary list) is not possible. However, a simple worst case analysis for high area-to-perimeter ratio regions can provide insight into the storage costs of hierarchical, regular areal representations.

A square of (n x n) pixels is a simple high area-to-perimeter (the circle possesses the maximum area-to-perimeter ratio) region for developing quadtree memory requirement bounds. The worst case (i.e., largest node count) occurs when the number of required low level nodes is maximized. A simple recursive algorithm develops a worst case node-assignment strategy by assuming that the entire perimeter of the square must be represented by the lowest level quadtree nodes. After removal of the lowest order border pixels of the square, the border of the remaining region is assumed to require representation of all next higher order quadtree nodes. This strategy is recursively applied until all n^2 pixels have been covered.

The algorithm requires first solving for the set $\{a_j\}$ that satisfies:

$$\Sigma a_j 2^j = n$$

where n is the pixel length of one side of the original square region, $a_j = 0$, 1 or 2 and j = 0, 1, 2, ... The number of nodes along a single border of the remaining square region at each stage of the recursion is given by:

$$c_{j+1} = (c_j - a_j)/2$$

where $c_0 = n$. Thus, the number of perimeter cells at each stage of the recursion is

$$P_j = \begin{cases} 0 & \text{for } a_j = 0 \\ 2c_j - 1 & \text{for } a_j = 1 \\ 4c_j - 4 & \text{for } a_j = 2. \end{cases}$$

Summing over all P_j yields the worst case node count.

Figure 2 compares 1) the pixel area (n^2), 2) the length of the region perimeter (4n-4) and 3) the node count for the worst case quadtree representation for a square region. Thus, the worst case cost (in terms of storage requirements) for representation of both the boundary plus the interior area in a minimal region quadtree compares favorably with the cost of the pixel boundary alone. As the area-to-perimeter ratio is reduced (e.g., small regions, regions with convoluted boundaries), the relative cost of an explicit areal representation, as opposed to an implicit areal representation, goes down. In the limit as the area of the region approaches its perimeter (pixel area), the non-pointer based region quadtree and boundary list memory requirements are identical.

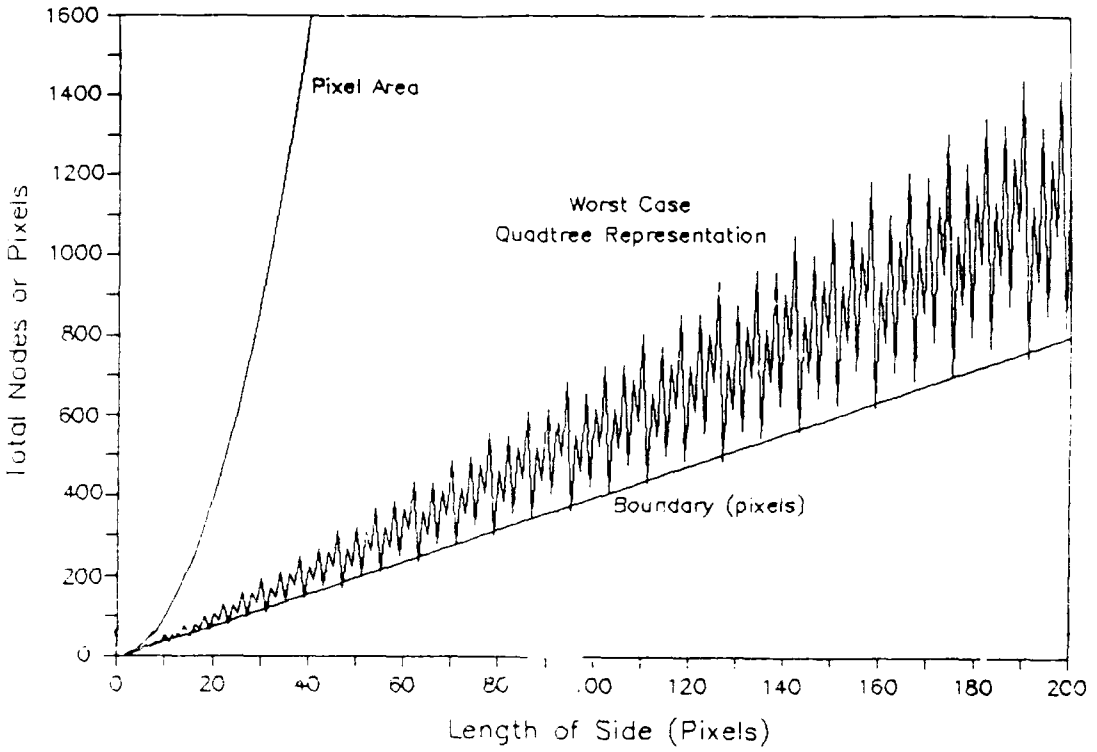


Figure 2 Worst case quadtree representation storage requirements for a square region compared to raster and boundary list representations (courtesy Cathy Lamanna).

In general, a spatial window is an arbitrary two-dimensional region that can be represented by a minimal quadtree. A window size can be scaled by a factor of 2^k (where k is an arbitrary positive or negative integer) by merely incrementing or decrementing subscripts in the quadtree locational code (node string notation) subscripts by k. The centroid (or any other node) of a window can be translated to an arbitrary quadtree node by calculating the directed distance between the current and desired centroid node location and applying the appropriate set of x and y offsets to all nodes within the window or by rebuilding the window about the new centroid position based on nearest neighbor node relationships within the original window.

Distance metrics can be developed between any of the six combinations of point, line and region features, such as closest distance, centroid separation and percent overlap. Since the quadtree is a relative coding scheme, the Euclidean directed-distance between two nodes can be computed as the term-by-term subtraction of the two node strings (or locational code). The x- and y- components of all possible resulting terms are summarized in Table 1. The relative distance magnitude between any two nodes is the square root of the sum of the squares of the respective x- and y-components.

Table 1 String interpretation for node evaluation and distance computation.

String term	x-component	y-component
A_i	2^{-i}	2^{-i}
B_i	-2^{-i}	2^{-i}
C_i	-2^{-i}	-2^{-i}
D_i	2^{-i}	-2^{-i}
$A_i - B_i$	2^{-i+1}	0
$A_i - C_i$	2^{-i+1}	2^{-i+1}
$A_i - D_i$	0	2^{-i+1}
$B_i - C_i$	0	2^{-i+1}
$B_i - D_i$	-2^{-i+1}	2^{-i+1}
$C_i - D_i$	-2^{-i+1}	0

Note: $(U_i - V_i) = -(V_i - U_i)$

Likewise, the x- and y-components of a series of nearest-neighbor moves can be accumulated so that arbitrary path lengths can be computed. The x- and y-components of each of the nearest-neighbor moves depends only on the node size and the move class (i.e., either cardinal or non-cardinal).

Using an image processing approach, both absolute and relative metrics can be computed between arbitrary combinations of point, line and region features. Since points and lines are merely degenerate regions, distance can be developed by incrementally growing regions until an intersection occurs. The efficiency of the algorithm can be improved by using hierarchical, multiple resolution region growth to rapidly locate and then refine candidate intersections.

2.3 Representation Issues

As mentioned in the Introduction, the key to memory efficiency is data compression. The key to search efficiency is storage of data along natural search dimensions. The key to problem solving efficiency is minimization of both the overall search space size and required transformation and manipulation of the data.

The following representation taxonomy for spatial features provides a natural first step in the top-down conceptual design of a spatial database:

1) True vs. non-true two-dimensional representation
2) Hierarchical vs. non-hierarchical representation
3) Regular vs. non-regular representation
4) Static vs. dynamic representation classes
5) Homogeneous vs. heterogeneous representations
6) Natural parallel and distributed database implementation potential

True Two-Dimensional vs. Non-Two Dimensional Representation

In general, a true two-dimensional representation inherently preserves adjacency, while in non-true two-dimensional representations, adjacency is either a catalogued relation or may require extensive search and test to evaluate. True two-dimensional representations store point, line and region features in a manner analogous to the acetate overlay paradigm. Thus, the region quadtree, pyramid and raster are considered true two-dimensional or explicit areal-based representations, while the polygon, boundary code, chain code and vector representations (which represent only region boundaries) are considered non-true two-dimensional representations. The underlying spatial decomposition can be either grid-based or non-grid based.

Set operations, spatial windowing, spatial search, distance metrics, moment calculation and spatial clustering are all inherently two-dimensional operations. Potentially large reductions in search space size and related generate-and-test operations are saved by supporting true two-dimensional spatial search of a database. Without the concept of adjacency of spatial features, the above operations may require search of large portions of the database. For a large number of regions, even the use of region bounding boxes (i.e., the set $(x_{max}, x_{min}, y_{max}, y_{min})$ for all features) may prove too costly. In addition, the bounding box is an ineffective search space control technique for extended line features such as roads, rivers and topographic contour lines.

Hierarchical vs. Non-Hierarchical Representations

Hierarchical representations support 1) problem solving at the most natural and efficient level of abstraction, 2) efficient top-down global problem solving and 3) generalization and specialization. Multiple resolution algorithms operate over the resolution continuum from global to local, potentially employing mixed top-down and bottom-up reasoning. For instance, the closest road to a particular (x,y) coordinate can be found efficiently by region-growing the point (x,y) in large step sizes until an intersection occurs, the intersection can then refined by higher resolution region contraction / growth. Likewise, a top-down recursive path planning algorithm can plan a route for an autonomous vehicle, while the same algorithm (operating at or near the maximum resolution of the database) can control the vehicle while enroute [Antony, 1986]. If an obstacle appears in the vehicle's path (e.g., a bridge is discovered to have been washed out), a path around the obstacle can be developed by reinitiation of the top-down path planning based on the present vehicle position and situation context.

Regular vs. Non-Regular Representation

Regular decompositions use a fixed grid size at each level of the decomposition and have a fixed resolution relationship between levels. When coupled with a true 2-D representation, regular decompositions support highly efficient spatial search, windowing and set operations. However, unless data is uniformly distributed at all resolution levels, regular representations tend to be highly memory-inefficient. In general, low resolution representations possess relatively uniformly distributed spatial features, while high resolution representations possess non-uniformly distributed spatial features. Thus, regular decompositions are most appropriate for low resolution representation.

Homogeneous vs. Heterogeneous Representation

Figure 3 depicts the data representation taxonomy for the above representation issues and Table 2 compares the relative efficiency of the first three representation attributes. In general, data structures that are non-hierarchical, non-true 2-D and non-regular are the most memory-efficient and the least search and problem-solving efficient. In effect, the compactness of the representation forces extensive search and "decoding" which, in the latter, are preserved at the expense of memory efficiency. Thus, an implicit tradeoff exists between the compactness of the representation and the efficiency with which the data can be accessed and utilized.

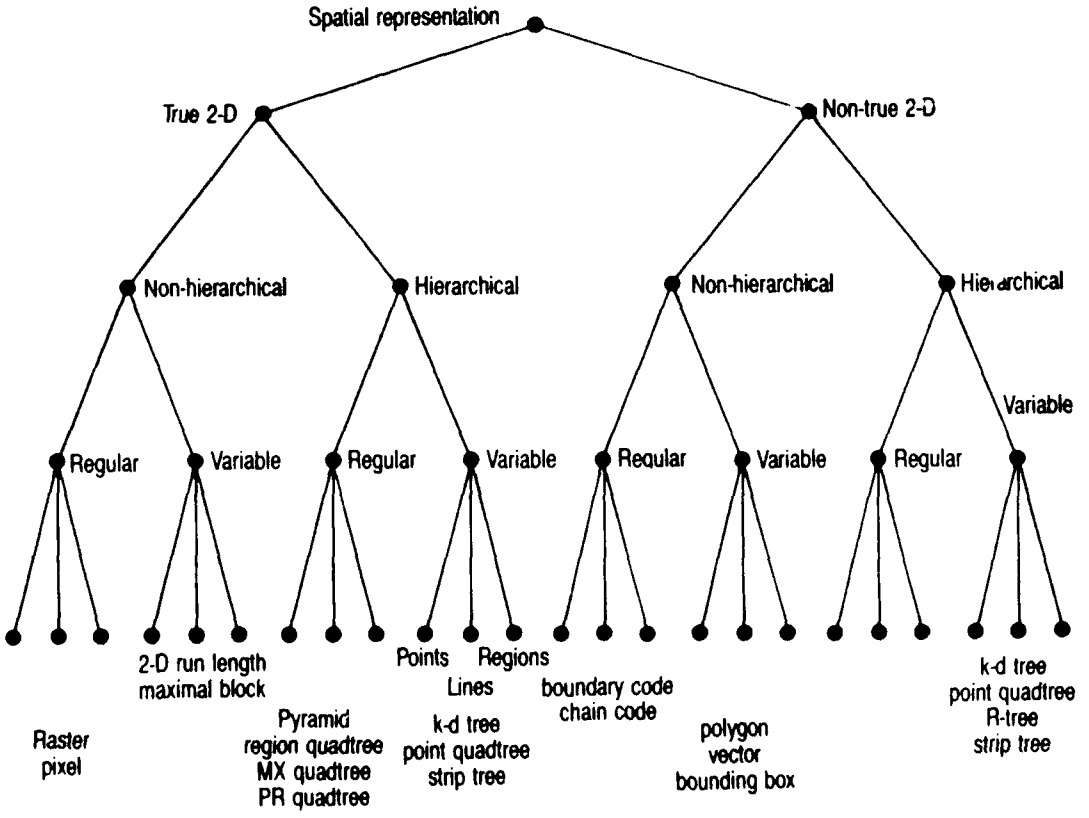


Figure 3 Spatial representation taxonomy.

Table 2 Relative comparison of memory, search and problem solving efficiency of various spatial feature representation classes.

Data Structure Classes	efficiency		
	memory	search	problem solving
True 2-D (explicit areal) Non-true 2-D (implicit areal)	X	X	X
Hierarchical Non-hierarchical	X	X	X
Regular decomposition Non-regular decomposition	X	X	X

While, conventional databases are often built upon a single primitive element (e.g., data record, linked-list structure), a hybrid approach to database representation and organization seems very natural since:

1) point, line and region features, as well as semantic objects and concepts must be represented;
2) a true 2-D areal representation is a highly efficient form for performing spatial search and set operations among all classes of spatial features;
3) hierarchical representations support the continuum from global top-down to highly local reasoning; and
4) regular decompositions provide efficient and powerful representations of uniformly distributed features; non-regular decompositions provide efficient representation of non-uniformly distributed features.

Static vs. Dynamic Representation Classes

Dynamic representations require relatively inexpensive insertion and deletion operations, while, static representations are those data structures that may require extensive tree rebalancing, table reconstruction or node splitting or collasping for insertion / deletion of data.

The temporal dependency of the data itself falls within three rough classes. Static data includes the relatively unchanging features such as elevation, soil type, roads, rivers and forest. Quasi-static features includes relatively slowly changing data such as weather, river crossing sites, situation and threat assessment. Dynamic data includes rapidly changing data such as sensor reports, current situation and collection management databases. Thus, in order to support efficient operations among all three classes of data, a database must possess relatively dynamic data structures.

Parallel vs. Distributed Database Implementation

Since the database is intended to support a broad range of applications over many levels of problem abstraction, a parallel and distributed implementation is highly desirable. Distribution of 1) data, 2) search and manipulation and 3) control across a processor network (either tightly or loosely coupled) effectively provides very large incore data store, while minimizing search and processing bottlenecks. A global database can thus be distributed spatially and/or logically within a network of nodes to satisfy demanding access, memory and processing speed requirements.

3. Proposed Database Framework

In general, semantic objects possess two-dimensional spatial attributes that can be treated as point, line and region objects. Based on the six database representation issues discussed above, a hybrid representation is proposed that provides an object-oriented representation of semantic objects, as well as an object-oriented representation of two-dimensional space. The representation consists of 1) a pyramidal multiple resolution, object-oriented spatial representation, 2, the region quadtree and 3) an object-oriented semantic representation. Figure 4 shows the essential components of the proposed spatial representation structure. Figure 5 depicts the composite view of the proposed database showing both the semantic and spatial object-oriented elements that coexist within a single object-oriented database (OODB).

At a low resolution, spatial data tends to be uniformly distributed; as the resolution is increased, such data tends to become non-uniformly distributed. Thus, an efficient database design should provide both uniform and non-uniform representations in support of low and high resolution reasoning, respectively. Low resolution spatial reasoning supports global decisionmaking by providing inherent focus-of-attention that restricts the size of the required search space prior to performing more refined (higher resolution) reasoning.

Only a non-regular representation can efficiently represent non-uniformly distributed high resolution point, line and region data. In the proposed representation, refined resolution data utilizes vectors to represent points and lines and minimal region quadtrees to represent regions.

The complete quadtree-based pyramid with its object-oriented cell representation provides a uniform, object-oriented spatial representation. The pyramid possesses an entity-relationship graph structure, where the nodes are object-oriented representations of spatial regions and the arcs define the geographic decomposition (and thus the parent/child spatial relationship between objects). General properties of features are inherited from below.

The point, line and region features that occur within each minimum resolution pyramid node will be represented as shown in Fig. 4. Point and line features can be efficiently stored as a hierarchical string name (e.g., class>subclass>...>name), an associated vector representation, and other optional attributes (e.g., time stamp).

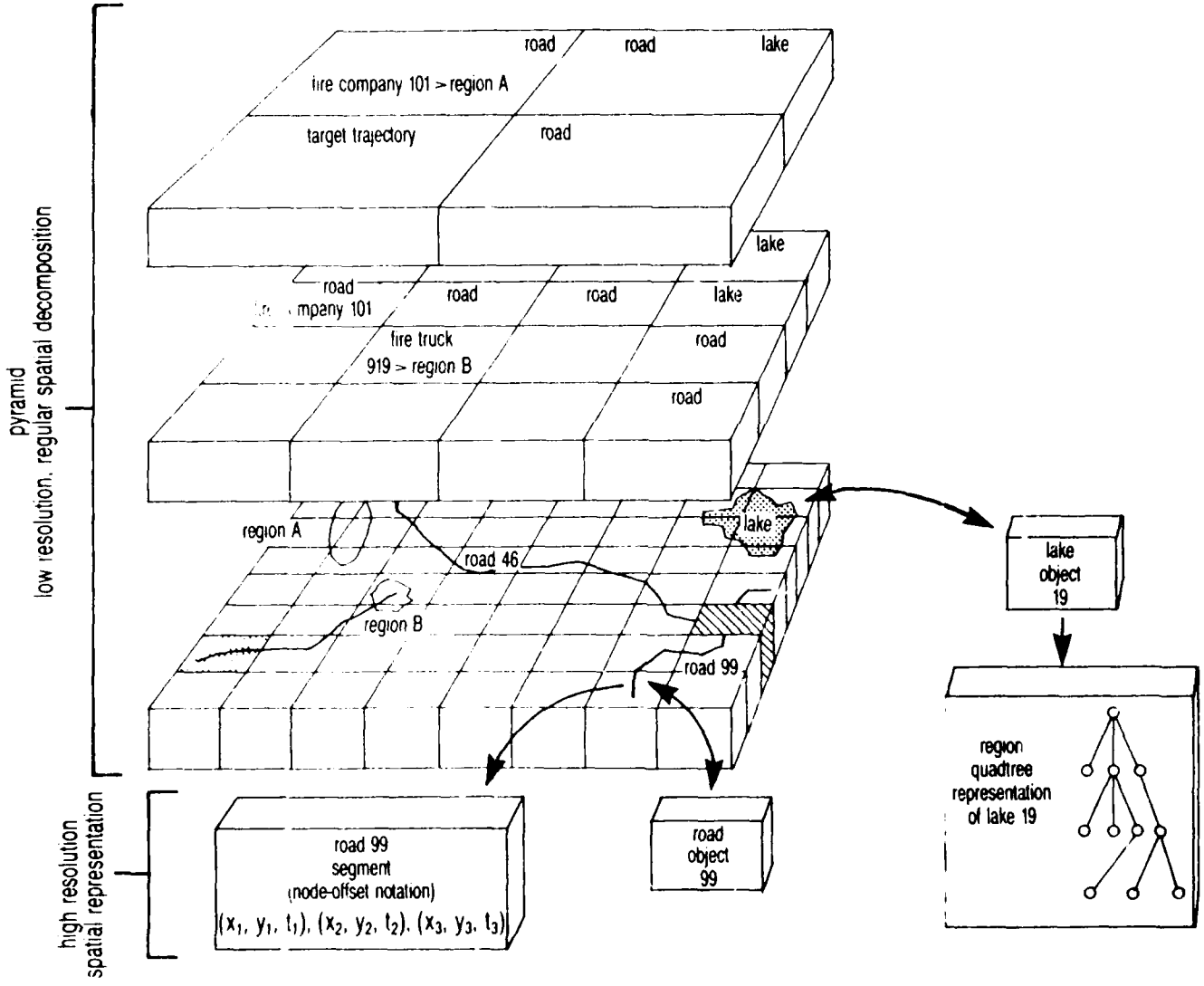


Figure 4 Hybrid low resolution and high resolution object-oriented spatial representation.

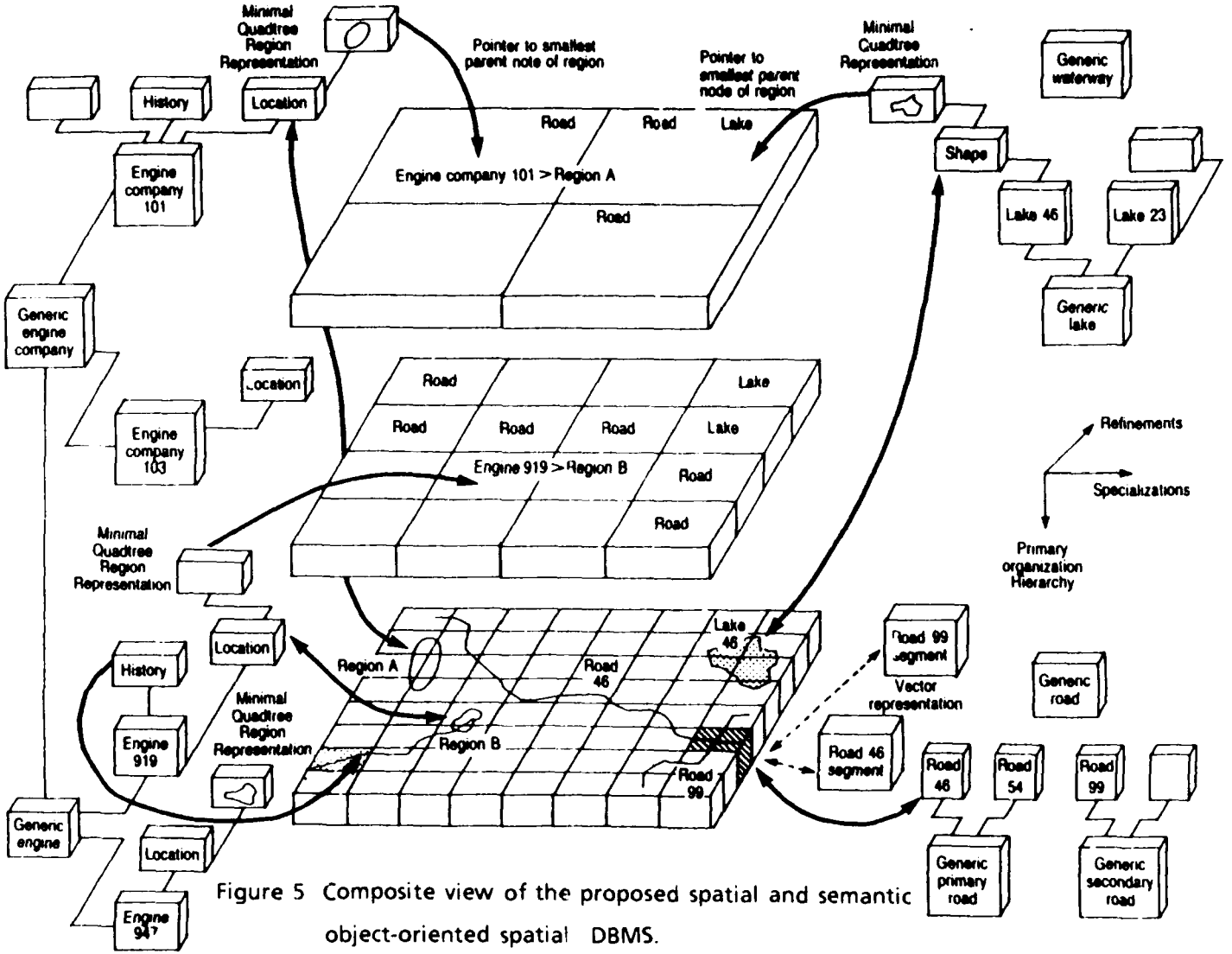


Figure 5 Composite view of the proposed spatial and semantic object-oriented spatial DBMS.

3.1 Point features

Point features within a minimum resolution pyramid cell have a straightforward mapping to the semantic object representation. Examples of point features include bridges, road intersections, buildings, and radar sites. For the inverse mapping, the semantic object frame points to a single minimum resolution pyramid node, which in turn maintains the precise location of the object in node-offset form. The node-offset form expresses x and y position offsets relative to the center of the respective quadtree node. For a 1 km minimum resolution pyramid cell, 1 m accuracy can be achieved by the use of 10 bit position offsets. Because of the location uncertainty and spatial extent of physical objects, certain "point" features may be more appropriately represented as regions.

3.2 Line features

Within a minimum resolution pyramid node, each line feature is represented as an object whose attributes are the vertices of a piecewise linear function stored in a quadtree node-offset form. Thus, the semantic line object consists of a set of spatially distributed objects each of which represents that part of the lineal feature that falls within a single minimum resolution pyramid cell.

Assuming a 100 km x 100 km region of interest, consisting of 100 roads with an average of 50 line segments per road, a raster representation of 10 m resolution requires 10^8 resolution cells, a pure vector representation requires $5x10^3$ end point pairs and the proposed representation requires 133 pyramid cells (for a 1 km minimum resolution pyramid node size) plus approximately $5x10^3$ end point pairs. The advantage of the latter representation is that for a large class of spatial reasoning problems, spatially conditioned semantic queries can significantly reduce the total number of raster cells or road segments that need to be searched.

For lines and regions, the relationship between the spatial and semantic object-oriented representations are not 1:1. An explicit pointer links the spatial representation of the named line segment to a specific semantic object. The inverse link points from the semantic object to one or more pyramid nodes that contain the line feature (e.g., line centroid, beginning node, all nodes). Any part of the line feature can then be accessed by simple nearest neighbor quadtree moves within the minimum resolution pyramid cells and use of the underlying vector data.

Since, in general, a search region has an arbitrary spatial shape (i.e., spatial window) and within such a window, set operations among point, line and region data may be required, the proposed hybrid pyramid/vector representation (with its uniform low resolution representation) offers advantages (e.g., supports efficient set operations) over strictly non-uniform lineal representations. In addition, with many standard tree-structured lineal representations, multiple intersecting lines present difficulties and large tree depths are required if line segment end points are represented at the pixel level.

3.3 Region features

The pyramid provides a hierarchical spatial indexing scheme for generic regions. Each pyramid cell (or one of its offspring nodes) point to the named semantic object for the regions present within that cell. In the inverse mapping, the region objects point to the minimal quadtree representation of the region. Since the root node of the quadtree representation (or some higher level parent node) has the same string representation as the pyramid node, the region object points indirectly to the pyramid representation. Thus, the pyramid provides a representation that facilitates hierarchical, multiple resolution region location and manipulation; refined region processing utilizes the respective minimal quadtree representation.

Although the "boundary-only" representations (e.g., polygon, chain code, boundary code) are very popular, memory-efficient region representations, since they are not true spatial representations, they do not support computationally efficient set operations (e.g., spatial windowing, intersection), metric computation or path planning. In addition, certain classes of regions (e.g., bounded regions with included holes, multiply-embedded regions) are not easily represented and manipulated.

3.4 Summary

A hierarchical, object-oriented spatial representation for point, line and region features has been proposed. When this representation is fully integrated with a semantic object representation, the object-oriented spatial DBMS (SDBMS) structure shown in Fig. 5 results. The primary organizational hierarchy is shown along the y-axis. In general, specialized objects inherit characteristics and attributes of their generic object class. The specialization of objects within a single class of the hierarchy is shown along the x-axis; further refinements are depicted along the z-axis. The pyramid provides a uniform, hierarchical representation of attributes that supports efficient spatially indexed set operations and associative processing.

Four natural frame-based object representation forms are used in the proposed database. The general *semantic object* has a conventional attribute-slot form with associated pointers and procedural constructs. The *pyramid object* employs either a general object structure or a simple bit-coded feature vector representation that summarizes point, line and region features found within its spatial bounds. The *vector object* is simply a link-list representation (in node-offset form) of named points or piecewise continuous line features. Finally, a *region object* is a minimal quadtree node representation of a named two-dimensional region.

The proposed SDBMS organization allows complex object-oriented queries, as well as efficient two-stage spatial reasoning. In the first stage of spatial reasoning, point, line and region features share a uniform, fully registered low resolution object-oriented pyramid representation that facilitates global reasoning, spatial focus-of-attention and offers potentially massive search space reduction over non-hierarchical, non-true spatial representations. Point objects are linked directly to the corresponding minimum resolution pyramid cell. The entity-to-spatial line object link allows continuous line feature-following via nearest neighbor moves within the pyramid. The region feature link is a pointer to the minimal quadtree representation of that region. The spatial object-to-semantic object

representation link for points, lines and regions is an explicit pointer from the spatial object to the appropriate semantic object. Thus, the proposed SDBMS provides both an efficient representation of two-dimensional spatial attributes of objects and features, as well as a natural spatial window into the semantic database.

The development of a generic spatial reasoning front-end for the proposed SDBMS would effectively create an intelligent database. The availability of such an intelligent database would simplify and expedite the development of application software, reduce the application-level communication requirements with the database and insure the efficient use of the rich object and spatial oriented SDBMS data structures. Thus, while the low-level SDBMS would handle primitives such as locking, data maintenance, data distribution and search, an intelligent database frontend would reformulate high level data queries into low level database operations, perform generic reasoning, and insure efficient search due to its intimate knowledge of the database. When desired functionality is not supported, the high level interface would simply be bypassed .

The intelligent interface can actually be implemented as a collection of specialized interfaces to a single (possibly distributed) SDBMS, each optimized for specific classes of operations or operational domains (e.g., target tracking, classification, multisensor correlation, generalization / specialization, path planning); likewise, the interface language can be tailored to the classes of generic reasoning performed by the database frontend. A more complete discussion of the details of the proposed SDBMS is found in [*Antony, 1989*].

4. Summary and Conclusions

A wide range of complex, real world manual decisionmaking tasks capitalize on the human's spatial reasoning ability. An automated system capable of performing real time, dynamic spatial reasoning is thus a critical element in the automation of such analytical tasks. Efficient problem solving depends heavily on the efficiency of the knowledge base search process; search efficiency depends in large measure on the organization of the database. Thus, problem solving efficiency can be enhanced by the use of a database that provides search dimensions that are natural to the problem domain. A key to efficient search is to store data along natural query dimensions that minimizes the size of the search space and the requirement for computationally expensive generate-and-test operations

The goal of the top-down design of a DBMS is to select representations that simultaneously achieve 1) memory efficiency, 2) search efficiency and 3) problem solving efficiency. Six critical representation issues were discussed:

1) True vs. non-true two-dimensional representation
2) Hierarchical vs. non-hierarchical representation
3) Regular vs. non-regular representation
4) Static vs. dynamic representation
5) Homogeneous vs. heterogeneous representation
6) Parallel vs. distributed implementation potential

A multiple representation approach allows the three broad design goals to be satisfied by trading among strengths and weaknesses of various representations.

Since automated spatial reasoning involves both object-oriented and spatial-oriented reasoning, an efficient implementation benefits from a fully integrated spatially and semantically organized database. In addition, the intrinsic advantages of a two stage spatial representation and reasoning process was emphasized. In the first stage, a hierarchically organized, uniformly sampled (regular) spatial representation provides efficient search space reduction by supporting both hierarchical and spatially-windowed search. In the second stage, non-uniformly sampled spatial representations provide memory efficient, refined representations of point, line and region features.

An object-oriented quadtree-based pyramid representation integrated with vector, region quadtree and semantic object representations was recommended as a powerful and robust, spatial database management system (SDBMS) organization. Specific applications may benefit from further specialization of the underlying spatial representations (e.g., an oct-tree to complement or replace the region quadtree representation). The unique characteristic of the proposed SDBMS is the maintenance of a hybrid uniform/non-uniform multiple resolution spatial object representation within a conventional semantic object-oriented database.

High speed access to the knowledge base depends on both efficient search and effective implementation. In a conventional DBMS, with host-resident software and disk-resident data, data access time depends both on the search space size and the disk retrieval time. The proposed SDBMS supports reduction in the search space size; a multiple processor implementation can substantially reduce disk retrieval requirements, since a significant portion of the overall database could be resident within active memory of a multiple processor network. In addition to the speed advantage of RAM access relative to disk access, such an implementation offers the potential for sophisticated concurrent search and manipulation.

The development of an intelligent database interface was recommended that transforms high level queries and spatial reasoning tasks into lower level generic database queries and operations. Such a capability would simplify and expedite application development, minimize communication with the database, insure efficient database search and mask the complexity of the underlying database organization.

References

Antony, R., 1989: A Hybrid Spatial/Object-Oriented DBMS to Support Automated Spatial, Hierarchical and Temporal Reasoning. In Su-Shing Chen (Ed.), Advances in Spatial Reasoning, vol. 1, ABLEX Publishing Corporation, Norwood, NJ, (in printing).

Antony, R., 1987: Spatial Reasoning Using an Object-Oriented Spatial DBMS, 1987 AAAI-sponsored Workshop on Spatial Reasoning and Multisensor Fusion, Pheasant Run Lodge, St. Charles, IL.

Antony, R. and Emmerman, P., 1986: Spatial Reasoning and Knowledge Representation, Geographic Information Systems in Government, ed. B. Opitz, pages 795-814.

DIGITAL TOPOGRAPHIC DATA SUPPORT

R.B. Lambert, J.A. Messmore, B.G. Rose, J.R. Ackeret, and R.T. Joy

U.S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060-5546

ABSTRACT

Artificial Intelligence (AI) has the potential to greatly enhance the effectiveness and lethality of Arry systems employed on the future battlefield. Given the Army's role as a ground force, however, it is clear that terrain knowledge will be key to the successful integration of AI tools into the planning and execution of battlefield tasks.

It is the responsibility of the Concepts and Analysis Division (CAD) at the U.S. Army Engineer Topographic Laboratories to ensure that today's researchers supporting future Army systems are cognizant of both current and planned digital topographic data bases. Through careful management of the Army's digital topographic data requirements, costs associated with the capture and use of these data will be minimized.

1. SUPPORT FROM THE ENGINEER TOPOGRAPHIC LABORATORIES

Within the Engineer Topographic Laboratories (ETL), the Concepts and Analysis Division (CAD) serves as the Army's center of technical expertise for all military applications of digital topographic data (DTD). Developers considering the use of DTD for any system application should coordinate with CAD as early as possible in the development cycle of their system. CAD's functions include: (1) maintaining technical liaison with DoD, DA, DMA and private industry, (2) providing technical support to the Army RDT&E community, (3) serving as a technical resource for combat developers and the analysis community, (4) conducting technical reviews of requirements documentation and doctrinal publications and, (5) providing/evaluating prototype terrain data bases. Through these functions, CAD can ensure the materiel and combat developer of reasonable database support, prevent the writing of overstated DTD requirements and eliminate redundant contractor support.

2. SUPPORT FROM THE DEFENSE MAPPING AGENCY

2.1 CURRENT DATABASES

The Defense Mapping Agency (DMA) currently produces more than fifteen (15) digital MC&G products. These products have been produced to satisfy the validated requirements for specific systems within the DoD community. As such, most of these databases have limited or no utility outside their intended system application. However, the Digital Terrain Elevation Data (DTED) and Digital Feature Analysis Data (DFAD) products are used for a wide range of military applications as general purpose, medium resolution MC&G databases. When referenced together these products are identified as the Digital Landmass System (DLMS). Each of these data bases has its own product specification.

2.1.1 DTED

DTED is currently available as a Level 1 standard product and consists of a uniform matrix of terrain elevation values with 100 m (3 arc sec) post point spacing. The information content is approximately equivalent to the contour information represented on a 1:250,000 scale map. The standard file size is a 1 deg x 1 deg geographical cell. DTED Level 1 coverage is available for many geographic regions worldwide and is distributed on 9-track magnetic tape.

2.1.2 DFAD

DFAD is currently available as a Level 1 standard product. DFAD consists of selected man-made and natural planimetric features, type classified as point, line or area features as a function of their size and composition. The data is stored in polygon format and segregated into 1 deg x 1 deg geographic cells. The information content is approximately equivalent to those features found on a 1:250,000 scale map. Both 1st and 2nd edition DFAD Level 1 products are available. The 2nd edition product contains lines of communication (LOC) data. The 1st edition data is available for many geographic regions worldwide, whereas 2nd edition data coverage is very limited. DFAD Level 1C, produced from map source, is also available for limited areas.

2.2 PROTOTYPE DATA SETS AND FUTURE DATABASES

2.2.1 DTED on CD-ROM

During FY89, DMA plans to begin distribution of DTED Level 1 on a compact, read-only memory optical disk (CD-

ROM). A key advantage of the CD-ROM over magnetic tape is that a CD-ROM cannot be written over or modified, thereby protecting the data from alteration. CD-ROM is a non-magnetic data storage medium and, therefore, is not affected by electro-magnetic pulse (EMP). Each disk has a storage capacity of 600MB and can hold well over 200 (1 deg x 1 deg) DTED Level 1 cells. A CD-ROM reader (disk drive) can be interfaced easily with a microcomputer, thereby providing a wide range of users with access to a terrain elevation data base.

Army is currently evaluating two (2) prototype data sets on CD-ROM that have been produced by DMA. One disk contains 341 DTED Level 1 cells and the other disk contains 275 DTED Level 1 cells, with coverage for Great Britain, Central and Eastern Europe and the Soviet Union. An applications software package, provided on floppy disk, enables the prototype evaluators to extract DTED information for a single coordinate location as well as the surrounding elevation posts, and perform data manipulations such as line-of-sight, tinted elevations, contouring and perspective views. The goals of the evaluation are twofold: to determine the advantages that CD-ROM has over 9-track magnetic tape and to determine the utility of DTED on CD-ROM for Army users.

2.2.2 DFAD Level 1C Plus

A new DMA prototype, DFAD Level 1C Plus, is likely to be released as DFAD Level 3C in the future. This 1:250,000 scale product contains 1:50,000 scale patches that are compiled from the 1:50,000 Topographic Line Map (TLM50). Consult the latest DMA catalog for availability.

2.2.3 Tactical Terrain Data (TTD)

TTD is intended to be the basic operational terrain data set supporting future land combat, close air support, and amphibious operations. TTD consists of both elevation and feature data. The elevation data consists of DTED Level 2 (1 arc-sec spacing). The feature data are consistent with the Tactical Terrain Analysis Data Base (TTADB) thematic overlays and special features from the 1:50,000 Topographic Line Map (TLM) and combat chart. A TTD prototype has been produced for a 15 min x 15 min area covering Fort Hood, TX. This prototype data set is available in both a non-integrated version (eight topologically separate theme files) and an integrated version (nine 5 min x 5 min tiles each with a single topological layer). User defined geographic information systems (GIS) must have the capability to extract user themes from the integrated version.

The elevation data is represented as a matrix structure; the feature data uses Minimally Redundant Topology (MINITOPO) as its structure. Feature categories will be those contained in the Feature Attribute Coding Standard (FACS). The exchange format for the initial TTD prototype is the Spatial Data Transfer Specification (SDTS). SDTS incorporates the American National Standards Institute (ANSI) ISO-8211 at the transport level.

Production of TTD is not scheduled to begin until after 1992. ODCSINT has recommended to DMA that initial production of TTD correspond to areas where extensive terrain analyses have been done, e.g., areas where TTADB's have been produced. While DMA has not endorsed this approach in writing, there are strong indications that DMA will adopt this production strategy. The determination of area requirements falls under the purview of the Commander-in-Chiefs (CINCs) of the Unified and Specified Commands (U&S Commands) that meet biannually with DMA to review area requirements. DMA then prioritizes those requirements in their Program Objective Memorandum (POM). Generally, strategic requirements have priority over tactical requirements. To our knowledge, actual TTD production plans have not been formulated by DMA.

2.2.4 Interim Data Bases

2.2.4.1 Interim Terrain Data (ITD)

ITD has emerged as the solution to providing a tactical-level digital product that will support the Army's near-term (1989-1993+) tactical and analysis community digital topographic data (DTD) requirements in the years before Tactical Terrain Data (TTD) becomes available in volume. ITD will be produced by DMA. The driver behind the new DMA ITD production program is the Army's Digital Topographic Support System (DTSS), which has a validated requirement for ITD. DMA will support the DTSS program with ITD, initially for upcoming system testing in 1989 and, eventually, for fielding between 1992-1994.

DMA proposes building ITD data sets initially through software conversion of existing Terrain Analysis Production System (TAPS) data, then by digitizing hardcopy (analog) tactical/planning terrain analysis data bases (T/PTADB's) and finally by new product generation via the DMA Mk.85 Feature Extraction Segment (FE/S) using data collection software designed for terrain analysis. By the end of 1988, DMA expects to produce the first ITD data sets through a conversion of the six 15 min x 15 min cells of DTD generated from the [now-defunct] TAPS. The

TAPS software conversion process will include both data exchange format and coding schema changes.

The majority of the ITD data sets will be produced from the existing analog products and will consist of six segregated files that represent the T/PTADB terrain feature data themes of surface drainage, surface materials (soils), surface configuration (slope), vegetation, transportation, and obstacles. ITD data sets generated on the FE/S will have integrated files rather than segregated files and will not contain slope data. For all ITD data sets, terrain elevation data will likely be provided as DMA Digital Terrain Elevation Data DTED Level 1 (3 arc-sec data). The ITD product exchange format is currently being negotiated; however, it will likely be DMA's Standard Linear Format (SLF), while the feature and attribute coding schema will likely be the DMA Feature File (DMAFF).

DMA delivered their draft Prototype Product Specifications for Interim Terrain Data (ITD) to the Army in September 1988. CAD has evaluated this document and is presently investigating other ITD and ITD-related issues including Army co-producibility of ITD, data structure, format and coding schema conversions, and quality control software development.

2.2.4.2 Electronic Map Displays (EMD)

EMD refers to digitally recorded, or scanned, pictures of maps that can be used to recreate the image of the paper map on a computer display. The Army articulated its requirement for an EMD product in the Statement of Requirements for a U.S. Army Electronic Map Data (EMD) Product. This document was forwarded by CAD to the Office of the Deputy Chief of Staff for Intelligence (ODCSINT) on 22 Jan 88. While no product currently exists that fully satisfies the Army's EMD requirements, DMA expects to produce a raster-scanned map product, ARC Digitized Raster Graphics (ADRG), in FY89.

The specifications for the ADRG format and storage medium (CD-ROM) were driven by the Navy's validated requirement for the product in support of the AV-8B Harrier Program. The ADRG is produced from hardcopy graphic products as a multicolor digital replica and does not contain separate feature files. The ADRG can, however, be reformatted to serve as a short-term interim solution to meeting some Army EMD requirements. A fully satisfactory EMD product would allow the Army user to manipulate specific groups of map features independently, as separate files, and would have minimal requirements for storage and pre-processing. CAD is investigating the

possibility of preparing an EMD prototype that will meet these requirements.

3. DMA DIGITAL DATA POLICY

3.1 DIGITAL MC&G DATA REQUIREMENTS FOR EMERGING SYSTEMS

Each of the Military Departments will: "Review ongoing system development programs to identify the need for unique mapping, charting and geodesy (MC&G) products. Beginning in the FY88 POM, fund [unique] MC&G activities in the program element for the associated system for later transfer to DMA for execution. Ensure that, as new systems enter full-scale development, the necessary funds for MC&G requirements are identified and programmed."[1]

3.2 DMA DIGITAL DATA TRANSFORMATION

All new transformation processing requirements will be reviewed through Department channels and approved by the Assistant Secretary of Defense (C^3I) prior to DMA implementation. New requests should be forwarded through DMA to Assistant Secretary of Defense (C^3I). The burden of proof is on the Department to show the economic benefit to DoD, or the operational reason DMA transformation processing should be approved. A major consideration is to maintain interoperability of MC&G digital data among DoD systems to minimize duplication, redundancy, and costs of processing.

3.3 DMA DIGITAL DATA REQUESTS

3.3.1 Fielded Systems

Activities requiring available digital data for use in Department-approved fielded systems may forward their requests in accordance with applicable Military Department regulations directly to the DMA Combat Support Center.

3.3.2 Developing Systems

Activities requiring digital data for any other application, and especially for proposed, prototype, or developing systems, will forward their requests through the appropriate channels and in accordance with applicable Military Department regulations. All Army activities will submit digital data requests to CAD, USAETL, for evaluation and endorsement. Endorsed requests will be forwarded, first, to ODCSINT for

---

[1] DEPSECDEF Program Decision Memorandum, August 1985

validation and, then, to DMA for final processing and distribution. Service validation of a request at ODCSINT may take 1 week; the processing time at DMA is 6-8 weeks. These processing times are estimates for routine requests only.

Customers should note that the minimum pipeline production time required for data on new geographic areas is approximately two years, assuming a high JCS priority. When contractors require data to support other than fielded systems, they must submit a request through their government sponsor.

3.3.3 Non-standard Data Requests

Activities requiring non-standard digital data must agree to the terms of a conditional release memorandum which restricts their use to proof-of-concept analyses, modeling, and simulation studies. These data shall not be used as part of any operational data base nor used as a system design criterion. A copy of the memorandum will be forwarded to the requestor for endorsement. The subject data will not be shipped until a signed copy of the endorsement has been received in CAD.

SCALE-SPACE REPRESENTATIONS FOR FLEXIBLE AUTOMATED TERRAIN REASONING

David M. Keirsey
Jimmy Krozel
David W. Payton

Hughes Artificial Intelligence Center
23901 Calabasas Rd.
Calabasas, CA. 91302

ABSTRACT

Representing terrain at multiple levels of abstraction can help in the automated terrain reasoning task. Scale-space representations provide the capability to quickly search for features based on the level of abstraction necessary for particular terrain reasoning tasks. The representation retains the finest resolution of the data while progressing through greater levels of abstraction, yielding a unique fingerprint for the terrain. Reasoning about how features change over levels of abstraction provides valuable information about terrain formations. A graph representation has been developed in conjunction with the scale-space image of the terrain to aid in retrieval of feature information. An example of feature extraction for an observation mission is given.

1. INTRODUCTION

With the advent of semi-automated generation of digital terrain maps in conjunction with high speed digital processing, the automatic analysis of terrain for military missions, or "terrain reasoning," has become feasible. On the other hand, terrain reasoning is computationally demanding because it can involve a great deal of geometric computation, and the computation must be done during both the specification of a mission and its execution. To reduce demand on computational resources, it is envisioned that flexible pre-computed spatial representations will have a role in reducing the computational complexity of geometric reasoning. This paper proposes methods for using scale-space representations in the analysis of digital terrain data. In this paper we discuss the problem, describe scale-space representations, and then present an application which combines scale-space with other representations for a sample mission problem.

2. PROBLEM

Planning of a military mission is a hard problem. There are a number of factors involved in choosing a successful mission strategy. The important strategic factors are usually based on terrain features and are difficult to envision in detail. Since the results of analysis depend heavily on the current military situation, pre-computed, *a priori* information obtained from a map may be useless in determining a course of action. Any pre-computed representations used must be flexible enough so that they can quickly be transformed or searched to find the appropriate information for mission tasks at the time of their execution.

Speed and generality are two advantages to using pre-computed representations rather than the typically more computationally intensive algorithms that use raw data and give exact answers. For example, consider the task of finding good observation points. What constitutes a good observation point depends on a number of factors, the primary ones being: the location of friendly forces, the probable location of enemy forces, and the accessibility to the units that are tasked to make the observation. In addition, the locations of all the players in the situation is constantly changing, and therefore a static analysis quickly becomes inappropriate. It would be computationally prohibitive to perform the calculations for the possible observation points by performing visibility computations based on enemy positions. In addition, since the situation will change, these computations would have to be repeated frequently. On the other hand, observation points chosen by heuristic measures can be obtained quickly. Although heuristically chosen observation points may not be optimal for any specific situation, their generality may make them preferable when uncertainty and change are taken into account.

When looking at map elevation data, one can see various *features* which may be considered important for planning. From a contour line representation of elevation data, regions of local minima and maxima are easily extracted. These local extrema are useful for reasoning about strategic locations in mountainous terrain. Slope data can be segmented into regions of similar slope. High slope regions are typically intraversable, and are therefore considered an important feature. One can naturally extend the search for features from analyzing raw elevation data and slope data to investigating the first deriviative of slope, i.e., concavity data. Concavity information should easily distinguish features like saddle point regions, regions amenable to masking, and the texture of a region.

Elevation map data does not change with time, however, some features intrepreted from this data may change with respect to the level of *abstraction* for which the data is viewed. For example, when looking at a mountain range, one notes different features at various levels of abstraction: at a high level, dominant mountains; at a middle level, major rock formations; and at a low level, individual rocks For military missions, the level of abstraction required for viewing the data is typically determined by the level and scope of the planning task. High level planning may require reasoning about mountain ranges, thus requiring a view of the data from a high level of abstraction. Likewise, the lowest level of planning involves very local features such as hills and gullies, thus requiring a low level of map data abstraction.

3. SCALE-SPACE REPRESENTATIONS

Representing terrain at multiple levels of abstraction can help in the automated terrain reasoning task. Scale-space representations provide the capability to quickly search for features based on the level of abstraction necessary for particular terrain reasoning problems. 1-D scale-space representations have been shown to be useful in characterizing signals abstractly (Witkin,Terzopoulos, Kass 1987). Extension of the 1-D scale-space to a 2-D surface provides a representation that uniquely identifies important terrain features.

The method used in this paper for identifying features within elevation data is based on scale-space filtering (Witkin 1983). First, scale-space filtering is introduced by presenting the method as applied to a 1-dimensional signal. Then, the generalization of this technique will be presented for 2-dimensional signals -- precisely the form of elevation data.

3.1. 1D Scale-Space

Scale-Space filtering is a technique for analyzing a signal qualitatively. When interested in determining the qualities of a signal that are most interesting, the extent to which one identifies a feature as important is largely dependent on the scale at which one looks at the signal. For instance, when the signal of Figure 1 is considered, one may say at an abstract level that the signal is characterized by three important features: a hump at the beginning of the signal, a period of low magnitude, and a final long region of large magnitude. At a lower level of abstraction, one may note the rippling in the low magnitude region, and at yet a lower level, one may look closely at the signal and note a bit of noise in the signal. How does one proceed to determine the features of a signal at various levels of abstraction?

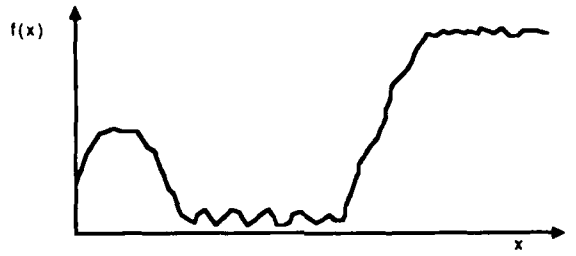


Figure 1. A general 1-dimensional signal.

Let us consider, in general, the method of scale-space filtering. A 1-dimensional signal may be smoothed by convolution with a Gaussian filter; smoother signals result from Gaussian filters with larger variances, as illustrated in Figure 2.

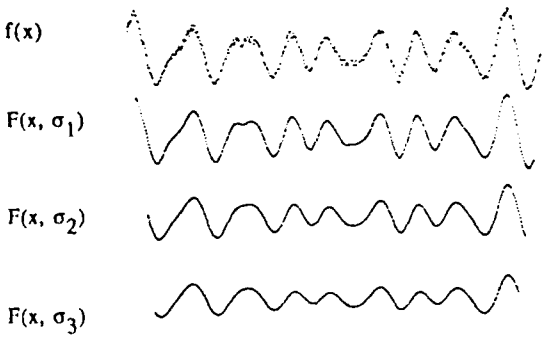


Figure 2. Smoothed Signals with varying Gaussian filters

Consider the variance of the Gaussian filter to be a parameter of *scale*. In scale-space filtering, one smooths the signal with increasing variances (scale) and records the locations of the inflection points of the resultant smoothed signals. These inflection points will move continuously upon continuous variation of the scale parameter. A plot of the inflection point locations versus the scale parameter provides a *scale-space image* of the signal (see Figure 3). The scale-space image is a unique description of the signal (Yuille and Poggio 1986) (Babaud et. al. 1986), thus one might consider this to be the "fingerprint" of the signal.

Figure 3. Scale Space Image

Analyzing the scale-space image of a signal establishes a qualitative interpretation of the signal. An individual contour within the scale-space image is termed an *event*; the geometry of these contours is surprisingly simple. An event originates at two distinct locations at the finest scale, and closes above at some *peak* scale. This property allows one to generate a coarse scale description of an event by tracing the event contour to its fine scale zero crossing locations. When one event terminates at a higher peak scale than another event, then we call the former more *persistent*. One can then describe an event in terms of its persistence through higher levels of abstraction.

While the persistence and inflection point root locations characterize an event, the events establishing important features within a signal will depend on the application. In Figure 3, the presence of noise in the signal is represented in scale space by the frequent occurrences of low peak events. The noise is quickly smoothed by the Gaussian filter, and thus does not persist through higher scales. We see that the mid-range of scales preserves most of the qualities of the signal. Finally, the highest scale eliminates all but the general form the signal.

The scale-space can be represented alternatively as a ternary-branching tree (Witkin 1983), which is a useful representation for searching for features through the levels of abstraction. Each node in the tree represents an *undistinguished* interval as σ is varied, where the interval is bounded by a pair of extremal points of opposite sign that appear in the smoothed signal. An undistinguished interval is where the extremal points do not contain any other extremal points. In general, each undistinguished interval is bounded on each side by the extremals that define it and bounded above by a singular point at which it merges into the enclosing interval and bounded below by a singular point at which it splits into three subintervals.

3.2. 2D Scale-Space

The application of the scale-space representation to two dimensional surfaces is straight forward. The gaussian filter in two dimensions is in the form:

$$F(x,y,\sigma) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(u,v)\frac{1}{\sigma\sqrt{2\pi}}\exp\left(\frac{(x-u)^2+(y-v)^2}{-2\sigma^2}\right)dudv$$

Figure 4 illustrates the difference between the original terrain data and a gaussian smoothing. It has been convenient, in the two dimensional case, to store a binary array representation of positive and negative concavity instead of storing the inflection points. This representation can be computed directly by convolving the Laplacian of a Gaussian filter with the original map. The scale-space image is a three-dimensional form defined by the zero-crossings in x,y,σ space. Successive applications of the filter yield more slices (see Figure 5) which can be stacked to produce a 3-D volume. This 3-D volume is the scale-space description of the terrain surface. A 3-D scale-space volume for a surface usually consists of numerous mounds, holes, caves, and tunnels. Each of these formations in scale-space reflects features of the terrain.

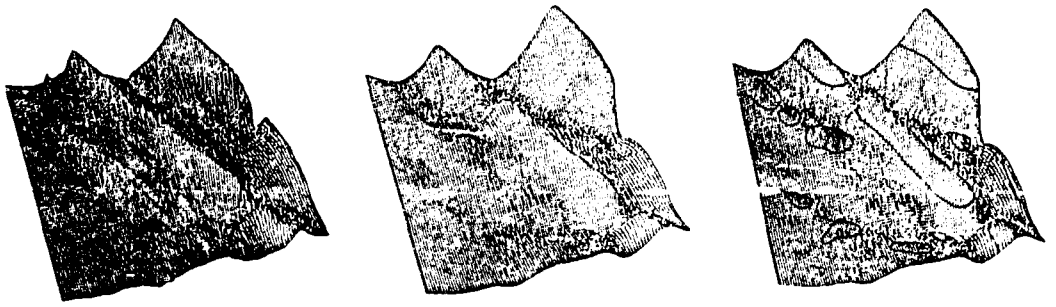


Figure 4. Original, Gaussian Smoothed, and Zero crossings of Terrain

As in the 1-D case, the 2-D scale-space representation retains the finest resolution of the data while progressing through greater levels of abstraction, yielding a unique fingerprint for the terrain. Reasoning about how features change over levels of abstraction provides valuable information about terrain formations. Regions that quickly disappear are small mounds or depressions and reflect only the nature of the terrain texture. Regions that persist and enlarge are large hills or long broad valleys. Various types of terrain formations have readily identifiable fingerprints.

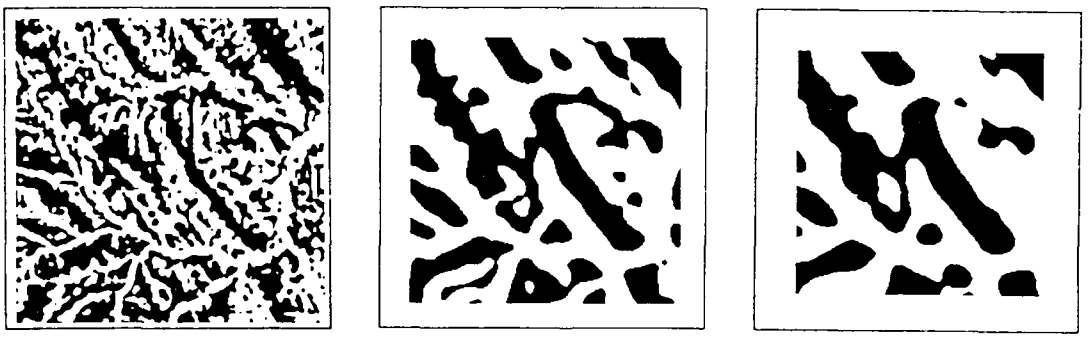


Figure 5. Slices of Scale-Space image

A saddle point, for example, occurs whenever two regions merge or split as one moves along the σ dimension of the 3-D scale-space volume. By tracking regions for each successive increasing σ, a graph can be constructed, indicating how the regions change (see figure 6). To find a saddle point, the graph is searched for merges and splits and the level at which the merge occurs indicates the width of the saddle point.

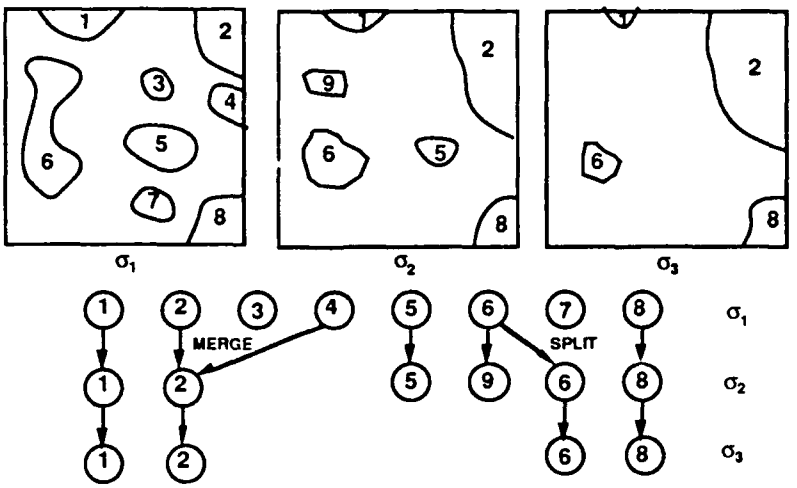


Figure 6. Graph Representation of Scale-Space

4. LOCATING OBSERVATION POINTS

The scale-space representation, in conjunction with other representations of the terrain, can be used extract important features. For example, consider observation missions. Good observation points are typically near or top of hills, and the scale-space representation can be used to quickly select the hills for possible observation points. Scale-space alone, however, is not sufficient for solving observation problems. There two other important terrain features for observation missions: accessibility and visibility.

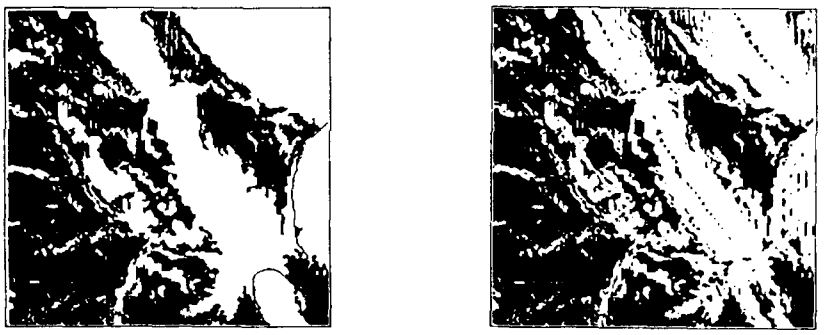
4.1. Accessibility

A general measure of accessibility for a given location is not easily quantified. Accessibility of a point from another specific location may be quantified by the cost of traversing the shortest route between those two points. Traversal cost, however, is dependent on mobility, and may vary significantly from one vehicle to another. A tracked vehicle, for example, may have greater mobility than a car in cross-country terrain, but a car can travel faster on highways.

Determining the accessibility of a location can be a computationally expensive task. Consider two methods for determining accessibility. The set of all routes found to a location determines a common set accessible points. Path planning using dynamic programming on an NxN grid representation can take a worst case N^2 search (Mitchell, Payton, Keirsey 1987). Although the worst case is the same complexity for heuristic search methods, these methods have a much better expected case complexity for determining accessibility. The second method for determining accessibility is to just determine connectivity. We first determine which points are traversable, then give all adjacent trabersable points the same label. Sets of points which all have the same label (connected components) form distinct regions in which all points are accessible to one another. Although this method does not produce a route between two locations, the advantage of this method is that it can be implemented in parallel, and the final representation can be used in accessibility queries.for other locations.

Figure 7. Terrain Test Area

We have implemented the connected components representation of accessibility and applied it to the digital terrain map using a model of mobility that is based on the vehicle characteristics of the DARPA Autonomous Land Vehicle (ALV). Figure 7 is an aerial photograph of the terrain used. Figure 8b shows the traversable areas of the test terrain. Figure 8a shows the accessible areas that are connected to roads.



(a) (b)

Figure 8. Cross-country terrain accessible (a) versus traversable (b) by the ALV

In military missions, areas that are connected by narrow corridors of traversable terrain (bottlenecks) have special significance. Figure 9 shows two significant accessibility bottlenecks in the terrain. An important problem is to find good locations for observing these bottlenecks.

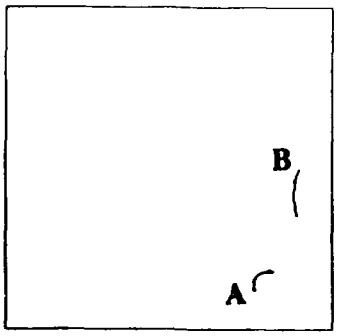


Figure 9. Significant bottlenecks in the terrain

4.2. Visibility

An observation mission involves finding accessible locations with good visibility to the area to be observed. Like accessibility, there are two types of visibility measures. The first measure, general visibility, is a measure of how well the location can be observed or how much one can observe from that location (both are equivalent measures). This measure can be quantified by the cumulative area of terrain that is observable from a given point. The second type, specific visibility of another area, can be quantified as the percentage of the area observed. It should be noted that visibility also depends upon the height of the observer (and if observing an object on the terrain, the height of that object).

Finding a good set of points from which to observe specific areas, such as those indicated in Figure 9, involves determining accessible points that have high visibility to those areas. A brute force method for finding all points that could observe an area would involve determining visibility for each point in the area. The time complexity would be $O(k \cdot r^2)$, where k is the number of points in the observed area and r is the number of pixels along the maximum range of visibility in the terrain map. The computation becomes prohibitive as the range or the number of points in the observed area becomes large.

As an alternative to computing the visible points explicitly, pre-computed measures can be used to provide more crude, but more readily available visibility results. As a heuristic for determining the best locations for observation, a visibility measure can be computed at each point. More specifically, let P be an observation point on the terrain grid, then the measure is a count of all visible pixels from P within a range R. This measure can be modified by adding a weight to each visible point based on the distance from observation point P. Figure 10 shows the visibility measure for part of the ALV test site, where lighter areas denote the most visible locations. Note that this measure does not retain information about intervisibility between specific pairs of points, but can be used as a general heuristic for finding good observation points.

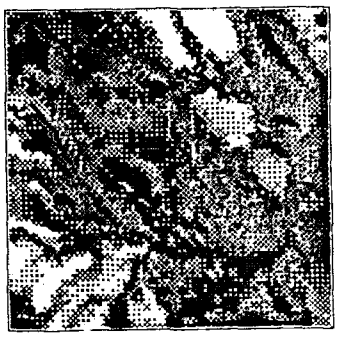


Figure 10. Visibility Heuristic Measure

4.3. Observation Mission

Several other features that indicate good visibility can be used to indicate the best candidate observation points. Along with the visibility heuristic, the slope of a candidate observation point can indicate the direction in which other points are most visible, (e.g., north facing terrain slopes can usually observe terrain to the north). For every point in the map, p, let **a**, be the projection of the local normal vector at p onto the horizontal plane. Given a centroid of a location to observe, c, an angle difference can be computed between the vector **pc** and the vector **a.** The magnitude of this difference is inversely proportional to the desirability for observation. This computation is easily implemented in parallel. Figure 11 illustrates the computation of the slope direction heuristic for observing a point near the lower right hand bottleneck, which we will refere to as bottleneck A. Darker areas denote more desirable slope direction for observing bottleneck A.

Figure 11. Slope direction heuristic

Using the scale-space representation, areas that are not good as observation points can quickly be eliminated by retaining only those candidate observation points that lie within the major concave downward regions (hills). This assumes that the good observation points are on hills. Figure 12 represents the hill regions for the map, extracted using a moderately abstract scale-space representation. The black areas indicate concave downward regions (hills).

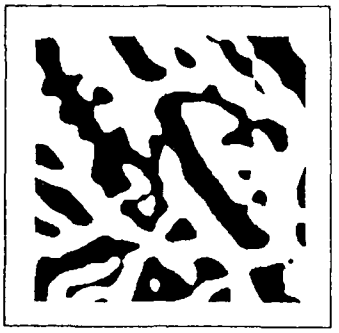


Figure 12. Moderately abstracted terrain

Candidate observable hills are chosen by performing a ray trace in all directions from the centroid of the area to be observed until a region in the appropriate scale is encountered. Combining the visibility heuristic, the slope direction, and the candidate concave downward regions yields candidate locations for observation. Intersecting the candidate observation locations with the accessibility map produces all accessible areas that have high potential for observing the bottleneck. The visibility heuristic and slope direction give a relative measure for the desirability of candidate locations. Figure 12 shows the candidate observation locations in the test site for observing bottleneck A.

Figure 12. Candidate observation locations for bottleneck A

5. CONCLUSION

We have found that no single resolution of data can serve as a basis for automating terrain reasoning. Abstraction of information without losing the important features is a key to any flexible representation. Scale-space abstraction arises from a spectrum of Gaussian smoothed terrain surfaces. Elevation, slope, and curvature information can be recorded for each smoothed surface; this information is then used to extract features for each level (scale) of abstraction. Reasoning about how features change over levels of abstraction provides valuable information about terrain formations. Various types of terrain formations, such as saddle points, have readily identifiable fingerprints. A graph representation has been developed in conjunction with the scale-space image of the terrain to aid in retrieval of feature information. Furthermore, scale-space representations have shown promise in quickly producing good qualitative results for mission planning problems that normally involve computationally expensive algorithms.

REFERENCES

Babaud, J , Witkin, A., Baudin, M., and Duda, R., "Uniqueness of the Gaussian Kernal for Scale-Space Filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* Vol. PAMI-8, No. 1, pp. 26-33, Jan., 1986.

Mitchell, J., Payton, D., and Keirsey, D., "Planning and Reasoning for Autonomous Vehicle Control," *Inter. Journal of Intelligent Systems*, Ronald R. Yager, Ed., Vol. 2, No. 2, pp. 129-198, Summer, 1987.

Witkin, A., "Scale-Space Filtering," *Inter. Joint Conf. On Artificial Intelligence*, Karlsruhe, W. Germany, pp. 1019-1022, Aug., 1983.

Witkin, A., Terzopoulos D., and Kass, M., "Signal Matching Through Scale Space," *International Journal of Computer Vision*, Vol 2, No.2, pp 133-144, 1987.

Yuille, A. and Poggio, T., "Scaling Theorems for Zero Crossings," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. PAMI-8, No. 1, pp. 15-25, Jan., 1986.

SPATIAL ANALYSIS FOR AUTOMATED TERRAIN REASONING

David L. Milgram, Richard F. Shu, Michael J. Black
Advanced Decision Systems
Mountain View, California 94043

1. INTRODUCTION

Terrain reasoning takes place amid a rich variety of knowledge sources. Among these are military doctrine, current situation descriptions, elevation maps, terrain features, weather and visibility conditions and reports. A typical terrain analysis will depend on information or models from more than one of these knowledge sources. Spatial Analysis for its part is not relegated to just one of these knowledge sources (e.g., the elevation map or the terrain feature map) but rather is concerned with the nature of the reasoning process.

Spatial Analysis refers to the reasoning methods which exploit the geometric properties of the knowledge sources (e.g., size, shape or placement attributes). Methods which compute nearness, width and intersection (to name a few) are spatial. Some properties of the terrain are not spatial; for example, the determination that the soil in a particular location will support a tank column after a rain storm is NOT a spatial analysis since there is no geometric component to the reasoning. However, computing that the particular muddy region constitutes a choke point IS a spatial analysis, since choke points are regions defined not just by intrinsic properties but primarily by their relationships to other adjacent regions.

Some spatial analyses are simpler than others because they rely on less of the potential complexity of spatial description. In general, we can identify a number of spatial complexity factors which provide semantic content. These include:

- Dimensionality of the space - e.g., 2-D, 2½ -D and 3-D.
- Scale - the size of objects considered significant (often identified with echelon: corps, regiment, company, vehicle).
- Dynamics - static behavior, sequences of events (e.g., route planning).

How much and which aspects of the complexity structure of space are required for a particular application is an important ingredient to a well thought out design.

2. PROBLEMS WITH SPATIAL ANALYSIS

The implementation of a spatial analysis application is likely to be judged according to two performance criteria: correctness and efficiency. Correctness is achieved when the formal algorithms designed to interpret concepts in the language of spatial reasoning do in fact compute results which correspond to the linguistic elements. Thus, for example, "containment" has a clear definition based on set theory; however, "nearness" is an informal concept of distance which is useful in many contexts, but whose definition is not that clear (Are the opposite sides of an abyss "near" one another?). How can the correctness of an algorithm for "nearness" be computed? In general, the correctness of an algorithm is judged formally (Does the algorithm exactly compute the definition?) and/or by experiment (Do people agree with the results of the algorithm for a reasonable set of tests?).

Efficiency is a measure of the utilization of resources available to the algorithm: time, mass storage, internal memory. Alternative implementations may differ according to the tradeoffs in their resource utilization. In terrain reasoning applications, the spatial analyses tend to account for a lion's share of the processing time required. It therefore makes sense to try to optimize the component spatial algorithms in advance of the application. Often, an algorithm optimized for one application is less than optimal for a different application. Deferring optimization until the application is about to run would be a good strategy.

3. ADS APPROACH

ADS has studied the problems of correctness and efficiency for spatial algorithms and has developed an approach based on the following principles:

- Structure terrain knowledge hierarchically.
- Use an object oriented approach to both algorithms and data.
- Provide highly interactive display and edit capability for spatial objects.
- Define a language of spatial operations and services.
- Hide details of representation and support conversion routines between them.
- Build statistical models to predict algorithm performance based on dataset complexity.
- Build a history file to record interactions and to explain how each region was determined.

4. ACKNOWLEDGEMENTS

This work was supported by the Spatial Data Structures for Robotic Vehicle Route Planning project sponsored by the U.S. Army Engineering Topographic Laboratories (ETL) under U.S. Government Contract No. DACA72-87-C-0015.

A MULTI-LEVEL KNOWLEDGE REPRESENTATION FOR REASONING ABOUT TERRAIN

Iris Cox Hayslip
Grumman Corporate Research Center
Bethpage, New York 11714

John F. Gilmore
Georgia Tech Research Institute
Atlanta, Georgia 30332

ABSTRACT

In this paper we present a three stage knowledge representation hierarchy addressing the question of the best way to represent complex tactical knowledge about low altitude air combat over hilly terrain. This research is part of continued development of Grumman's Rapid Expert Assessment to Counter Threats (REACT) system. The problem of integrating into REACT the capability to reason about the tactical use of terrain has produced results on knowledge base design and large database management, system and language specifications, and planner architectures pertinent to realtime coupled systems. It is unrealistic to assume that one can eventually make all modules of a complex AI system operate in realtime for a close air support domain. This aspect demands multi-level reasoning, motivated by the desire to use the best and deepest knowledge which can be made available in time to act. The REACT multi-layered representation is designed to support detailed planning, when time allows, as well as top level approximate solutions in time critical situations. The highest level of the hierarchy contains pre-compiled procedural knowledge for the most approximate, fastest solution. It is a structure including specially reduced terrain patterns with associated maneuver choices pre-stored for various relative opponent placements over the map. The middle layer of the hierarchy consists of terrain descriptions in terms of tactical terrain primitives providing a symbolic description for access by inference rules. The lowest layer contains DMA maps. This is the layer where the traditional (numerical) models live, although it is also possible within REACT to manipulate models at the low level using production rules. We discuss aspects of opportunistically choosing the level of terrain reasoning utilizing the various terrain representations as necessary.

1. INTRODUCTION

The Rapid Expert Assessment to Counter Threats (REACT) project addresses the critical problem of providing pilots with decision support in the complex situations which arise in high threat combat scenarios, in particular avoiding and countering threats in

dynamic battle environments. The problem of encoding pilot knowledge and reasoning processes has been addressed using Artificial Intelligence (AI) techniques. But the real-time requirement for this critical combat domain requires special AI architectures and a careful mix of AI techniques and conventional algorithms, pushing research well beyond the current state of expert systems technology. Primary focus is on providing realtime response in a realistic domain. Domain realism will eventually require modeling all aspects of the combat environment including terrain factors and other environmental factors, multiple cooperating threats, and domain uncertainties.

Recent research has led the project to focus on three major objectives. The first is the incorporation of advanced reactive planning methods into the REACT system. The second objective is the development of a hierarchy of knowledge representations which will maximize the amount and depth of knowledge which can be accessed in time critical situations. A third objective is to explore means for the REACT system to monitor its own performance and learn from its mistakes. This paper presents the state of research on the second objective, the knowledge representation hierarchy. Throughout this paper the use of the word 'hierarchy' refers to a hierarchy of abstraction in the data/knowledge base. The initial motivation was to provide a world model for REACT with the necessary information for reasoning about the tactical use of the terrain; analogous to the top level information the pilot would use if looking out at the terrain features in a combat situation. And when this representation cannot be utilized fast enough for time critical decisions, we seek to recognize the lack of time and employ quicker, approximate solutions.

Previous efforts on this problem for the ground battlefield scenario are reported in [McDermott and Gelsey 1987; Kuan 1984]. Fundamental differences in that domain and the air battle domain result from the real time aspects - the amount of time available to make decisions. However our above the ground domain can present an easier problem in one sense: more detail of the terrain can be ignored (more smoothing, coarser processing). This work extends previous results to include aspects necessary for the air battle over terrain, specifically concentrating on the reaction phase in planner execution as well as overall speed of database management.

The paper is structured as follows. Section 2 supplies the requisite description of the REACT system architecture. This section clarifies the functions and focus of various modules of the system, providing motivation for the knowledge representation hierarchy design. Section 3 discusses the realtime transitionability constraints which were primary motivators of the representation design. Section 4 presents the knowledge representation approach and defines the design schema. Section 5 explains how the high level terrain representation could be incorporated in a knowledge acquistion tool. Section 6 outlines implementation progress. Section 7 contains conclusions and directions for continued research including research in progress on

using other methods to generalize the system knowledge, such as using a neural net to store terrain pattern and response pairs.

2. OVERVIEW OF THE REACT SYSTEM

The top level design of REACT is a set of cooperating "expert systems" communicating by the use of a blackboard (Figure 1). Top level descriptions of the REACT system can be found in [Rosenking and Roth (1988)] and discussion of the blackboard implementation is found in [Roth et al. (1987)].

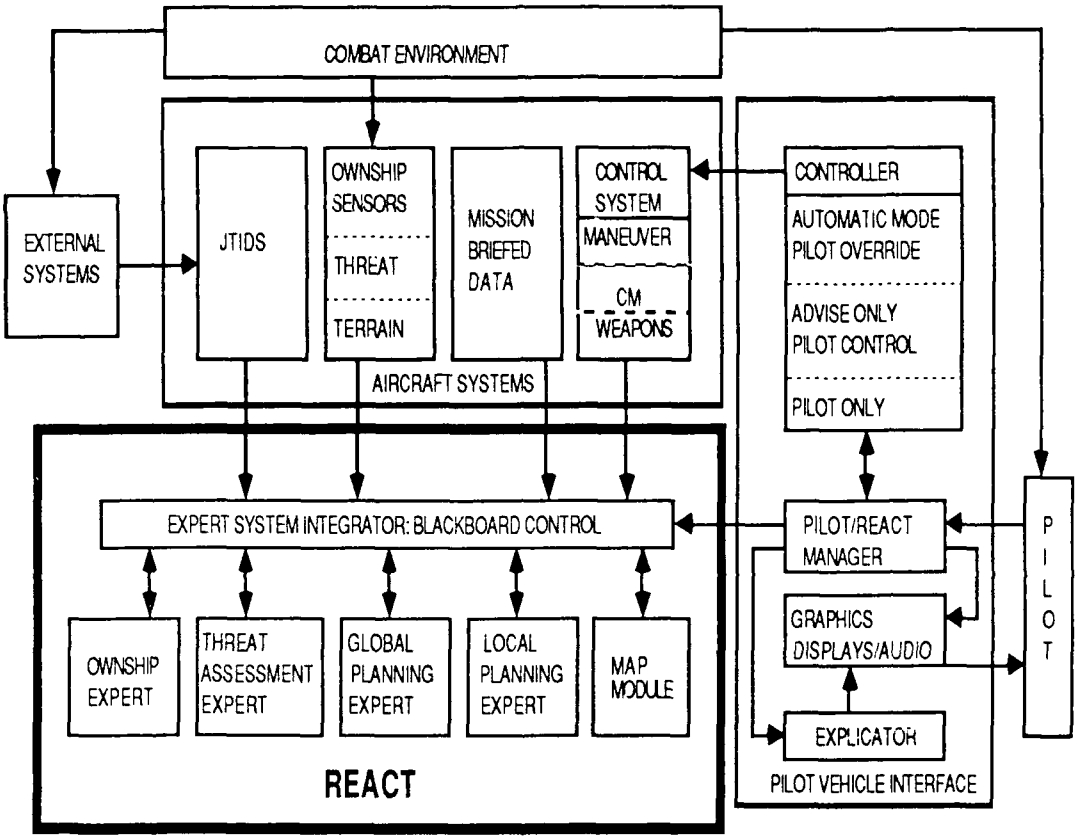


Figure 1: Initial REACT Architecture

The extended REACT architecture [Figure 2] consists of a blackboard with five cooperating systems . The GLOBAL PLANNER, based on the TREK route planner developed by Georgia Tech Research Institute [Gilmore and Semeco (1986)] provides global route planning and replanning. The THREAT ASSESSMENT expert will provide threat location and situation assessment. The OWNSHIP expert will provide status determination of the aircraft's systems. The MAP MODULE, a recent extension to the REACT top level design, is tasked with maintaining the REACT system's internal representation of the world, including the database and memory management associated with terrain maps. The LOCAL PLANNER determines the best local trajectories for fast threat avoidance. Current design has decomposed the LOCAL PLANNER into a system of cooperating modules

in a nested blackboard architecture.

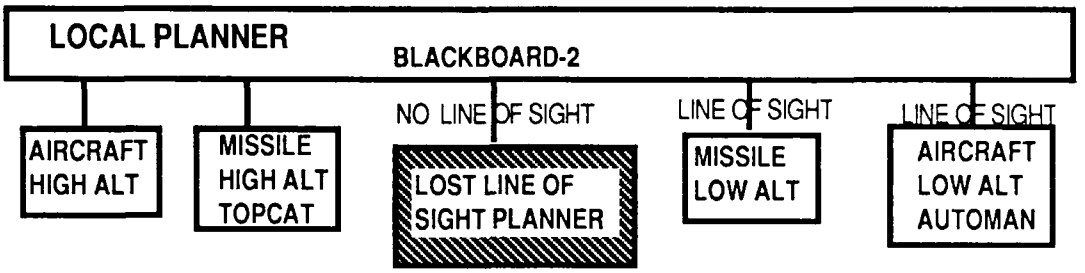


Figure 2: Extended REACT Architecture

Major effort has been and will continue to be concentrated on the LOCAL PLANNER component of REACT. The LOCAL PLANNER is decomposed into cooperating modules, some of which utilize conventional models embedded in a structure that allows the models to be controlled and adjusted by heuristics. The modules tasked with handling high altitude threats can utilize results from the Tactics Optimization and Combat Analysis Tool (TOPCAT) [Falco et al. (1974)]. TOPCAT is a program for determining optimal air combat maneuvers. It has been developed over fourteen years and has been used for effectiveness studies for various vehicles. It generates tables using a stochastic learning algorithm that permits air combat maneuvering strategies to be optimized by simulation of the combat using accurate aircraft models. Three submodules of the LOCAL PLANNER are tasked with determining maneuvers at low altitude in the presence of terrain. When the threat is within line of sight (i.e. not masked by terrain) scoring function techniques modeled after the Automated Maneuvering (AUTOMAN) [Austin et al. (1987)] system can be used. The AUTOMAN program is a realtime maneuvering program which has been installed and demonstrated in the NASA Ames Vertical Motion Simulator. We feel that the use of a scoring function tuned by heuristics is a good approach for real-time reaction to visible threats at low altitude. When line-of-sight to the threat is not available then the LOST LINE OF SIGHT (LLOS) submodule of the LOCAL PLANNER is in control, tasked to capture the heuristics, guesswork, and tactical maneuvering that a pilot would exercise in combat.

The LLOS module [Figure 3] contains the experimental planner resulting from REACT research on adaptive, reactive, realtime planning problems. It is at this level that the system uses multiple knowledge representations and several types of planning processes dictated by the time available to plan. The LLOS planner uses models, traditional knowledge representations, and precompiled procedural knowledge. The basic motivation of the architecture is to produce concurrently operating planners each operating with different levels of detailed knowledge and at different time scales. The planners compete for control, with the chosen plan being the best one which is available on time [Kaelbling (1986)]. Suboptimal plans are delayed by activation time deadlines. Each submodule of the LLOS planner can utilize the data/knowledge representation which is most suitable for its task and speed

allotment. In particular the QUICK RESPONSE module in the LLOS planner will access the top level terrain knowledge representation. The modules are designed so that this "racing" system will result in functional task subsumption [Brooks (1986)], that is, higher level behaviors will be executed whenever possible. Details on the REACT planner architecture are found in [Hayslip and Rosenking (1988)].

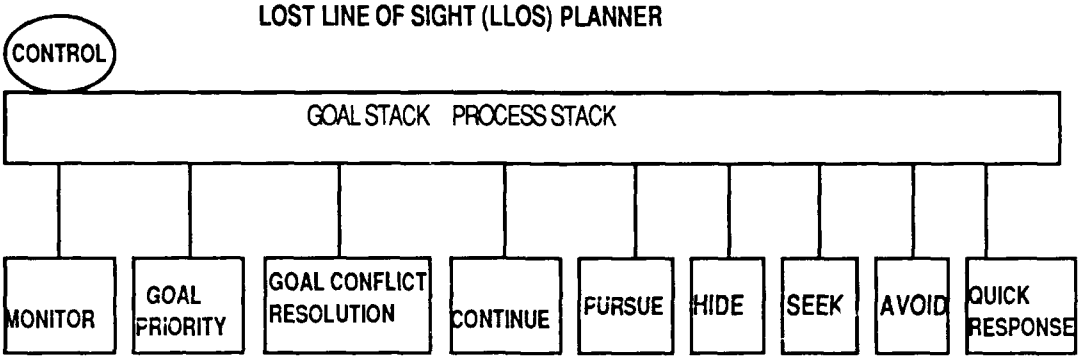


Figure 3: LLOS Planner Submodule of the Local Planner

3. REALTIME ARCHITECTURES

High level goals for knowledge representations suitable for a real time system can be outlined as follows: 1) Use as much knowledge as possible for a quick decision. Even if a reactive decision can only be "table look up" this means reducing the representation so that as much knowledge as possible is encoded in the data structure and that the structure provides adequate retrieval time. 2) More time to plan should mean more detailed or deeper knowledge can be utilized. This indicates a reasonable cost function or other method to assess time available for planning, and a hierarchy of abstraction in the representation. We seek a method for knowledge organization and distribution which facilitates the multi-level reasoning capability required by a realtime system. The hierarchical representation is used in order to not use details that won't affect the choice of maneuver. At one level (when there is some time available for "thinking") the world representation must be sufficiently reduced that a situation could be presented symbolically to an "automated pilot", and his choice of maneuver could be reached by reasoning with rules gathered from experts. At the highest level the maneuvers must be immediately available. In this case a fast look up is warranted. At this level the representation will resemble the TOPCAT [Falco et al. (1974)] results tables, except that the maneuvers are discovered through knowledge acquisition as opposed to simulation results.

In the LOCAL PLANNER, maneuvering is determined by the AUTOMAN [Austin et al. (1987)] program when the two aircraft are within line of sight. The AUTOMAN program uses DMA data directly. The development of the higher level terrain representations has been principally motivated by the lost line of sight problem. However, the representation hierarchy could also be used in the

opponent-in-view case. Notice that the pilot can lose sight of the opponent for reasons other than being masked by the terrain, i.e. clouds, position behind or under the aircraft, etc. The current concentration is on the problem of determining what to do when the opponent disappears behind a hill. Another simplifying assumption motivating the current implementation stage is that of mutually aggressive opponents. For example, the case of one pursuer and one evader, etc. is not currently being implemented. This assumption means that the opponent will be using the terrain to conceal, confuse, trick, etc. but not to escape. It is however possible that temporary retreat is a valid attack tactic.

4. APPROACH

First the DMA map containing the global path (mission path pre-determined by the GLOBAL PLANNER) and surrounding areas is divided into "rooms". In the case of an attack mission a sequence of rooms can be indexed by a hash table paramaterized by the distance to target along the attack path. Otherwise, an array of maps around a landmark in the battle area can be used. One of the tasks of the MAP MODULE in REACT would be to retrieve from disk the current and next "rooms" and their associated data structures. The MAP MODULE transfers the middle (symbolic) level representation into frames for access by the symbolic reasoning system. This module is also tasked with reallocating memory when the maps are no longer pertinent.

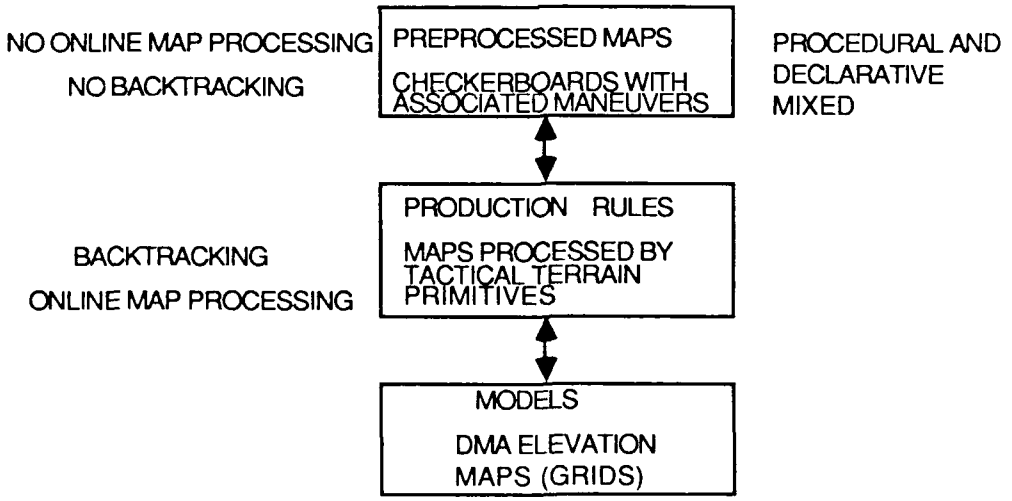


Figure 4: Knowledge Representation Hierarchy

The knowledge representation requires a suitably expressive world representation (for states), and a procedural knowledge representation (for state transitions). One could view one room map with two opponents positioned in the world as states, and tactical choices to move opponents from state to state as state transitions. We will describe the three ways to represent this knowledge that are being developed for the REACT system. The terrain is supplied as DMA maps. Clearly this structure is of the wrong form to use as an internal representation for symbolic

tactical reasoning, and it is too detailed for fast, approximate reasoning. However it still must be available for access by procedural attachments (models) for certain battle situations. Figure 4 depicts the basic stages of the representation hierarchy; the following sections will describe the structure and access conditions for each.

4.1 TOP LEVEL

Our "racetrack" architecture demands one module which is always ready. If more detailed, better plans are not ready in time, the system must produce its best guess. The top level of the representation hierarchy is used to produce the fastest, most approximate solution. It contains the data structure which is accessed by the QUICK RESPONSE reactive module in REACT. It is designed to be a database containing precompiled procedural knowledge [Georgeff and Lansky (1987)] and accessing the knowledge reduces to a database retrieval. The terrain database domain is an extremely demanding problem in terms of the tractibility of efficiently storing types of terrain patterns coupled with appropriate pre-stored maneuvers. In order to manage the amount of data needed to represent the terrain for the dogfight world the data must be reduced, and the database designed such that the retrieval time is adequate for quick response. Such a data reduction scheme is described below. The structure is derived from reduced terrain patterns with associated maneuver choices pre-stored for various relative opponent placements over the map. Local similarities, and ballpark threat placement, as well as matching and generalization schemes reduce the number of maps which need to be pre-processed. States in the representation are defined to be reduced map grids with ownship and opponent positioned on the map, rather like a checkerboard with two positioned players. State transitions are to be determined by database look up, where the maneuver choices in the database are assigned by pilot knowledge (see Section 4).

During system operation, the current and next (or neighboring) rooms with their associated data structures will be retrieved by the MAP MODULE. The rooms are subdivided and stored in quadtrees [Samet (1981)]. For each level in the quadtree, the map sectors will have been preprocessed into "checkerboards" with desired resolutions. The choice of desired resolution or mesh of each checkerboard is determined apriori by the roughness of terrain. If the terrain has little variability, then big squares are better.

The digital terrain map is transformed into a reduced pattern, or "checkerboard" representation. This is accomplished as a four phase process guided by predetermined mission parameters that are adaptive for a variety of applications. The first step determines the internal map quadrant size. As each terrain map is 600 by 512 pixels rather than square, the checkerboard is partitioned into n by m grid regions where the size indicates the lowest level of resolution desired for the current mission. Size may be a functional combination of terrain, threats, aircraft speed and

maneuverability, or mission goals appropriate to represent the current state space. For example, Figure 5b is a representation of the terrain map partitioned into 80 regions while Figure 5c exhibits a higher resolution partitioning into 320 regions.

Once the partitioning has been determined, the second step is to extract terrain features for each region to determine the associated region labels. These features include altitude variance, altitudinal differencing, mean altitude, gradient direction, and briefed ground threat exposure probability. The third step involves classifying individual regions on the basis of their feature space definition. Five primary terrain classifications exist (mountain, plateau, hills, flat area, and corridor) and are designated as follows.

The smoothness of the terrain within a region is analyzed by examining the variance across the region and the region's mean altitude and altitudinal difference. Predetermined system thresholds of high, medium, and low are used for comparative analysis of each feature. A high variance indicates possible mountainous or hilly regions. A medium mean with large altitudinal difference results in the label of mountain. A low mean with a low altitudinal difference produces a label of hilly. Low variance indicates the possibility of plateau or flat areas. High mean and altitudinal difference values translate into a plateau designation, while low mean and altitudinal differences identify flat areas such as fields. Corridors are a special case of mountain region wherein two distinct altitude differences are detected indicating a terrain corridor such as a valley region. Four secondary corridor designations (horizontal, vertical, rising left, falling left) exist based upon the gradient direction feature.

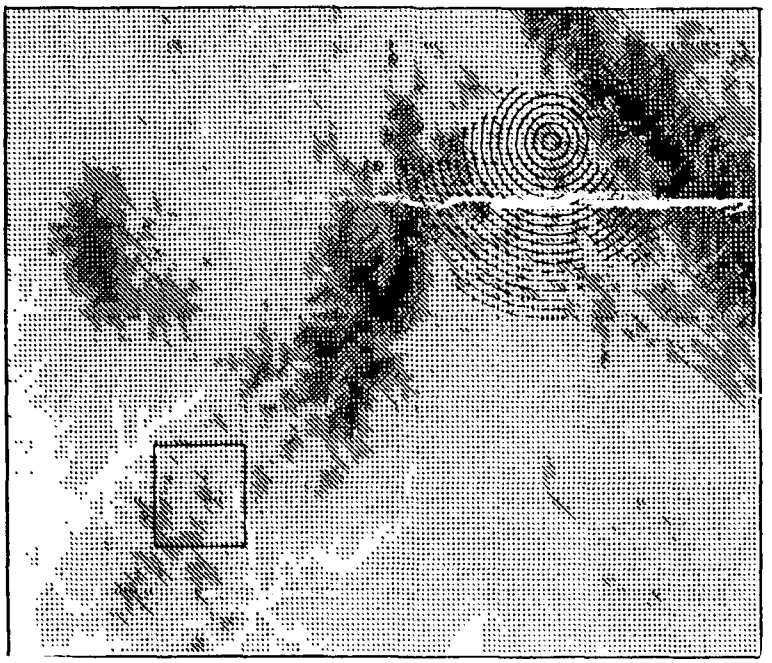
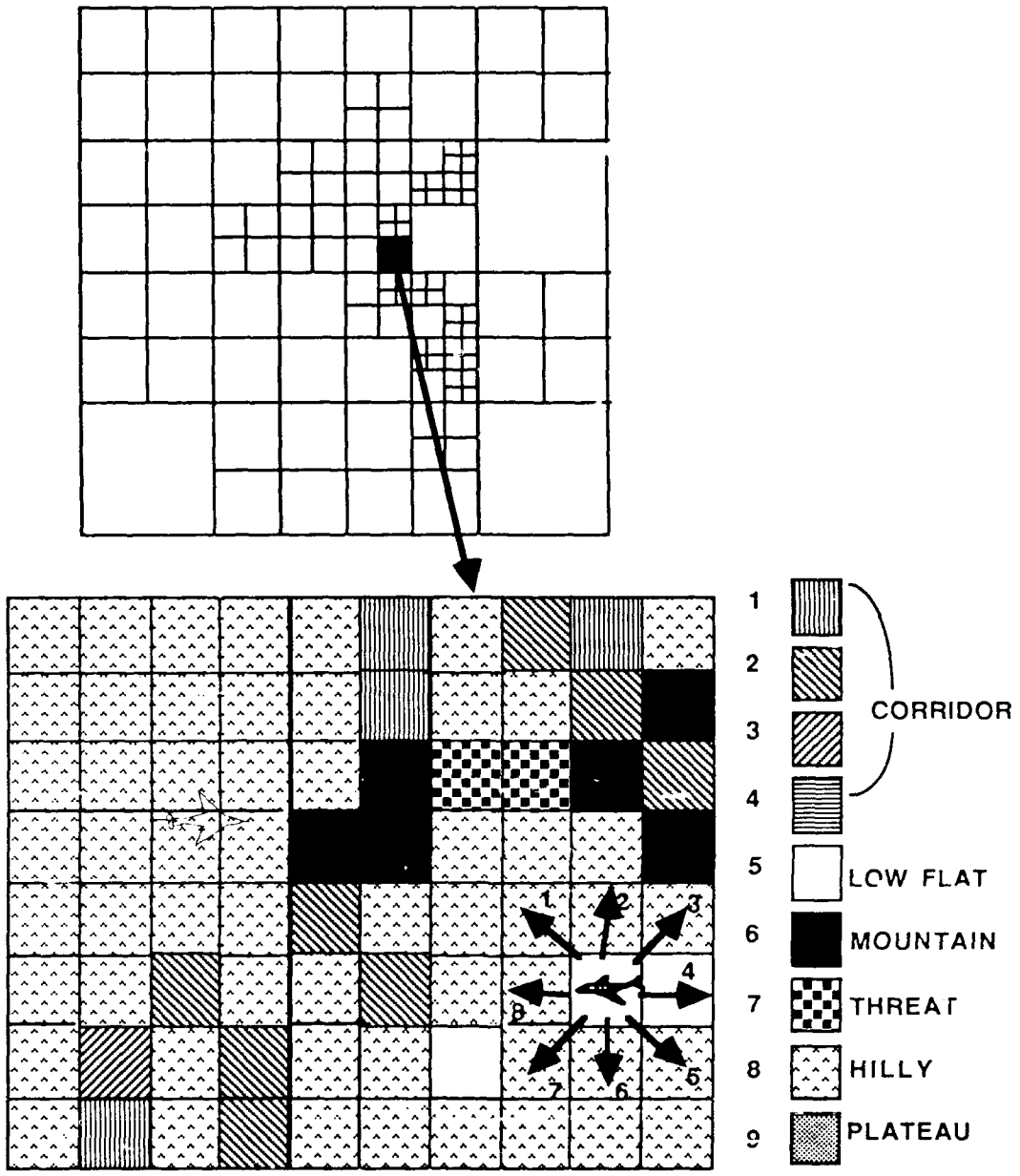


Figure 5a: Terrain Map used to produce Figures 5b and 5c.



MAP

8 8 8 8 8 1 8 2 1 8
8 8 8 8 8 1 8 8 2 6
8 8 8 8 8 6 7 7 6 2
8 8 8 8 8 6 7 7 6 2
8 8 8 8 2 8 8 8 8 8
8 8 2 8 8 2 8 8 5 5
8 3 8 2 8 8 5 8 8 8
8 1 8 2 8 8 8 8 8 8

RED (3,3) BLUE (6,9) MANEUVER 8

Figure 5b: A reduced terrain map, n=8,m=10.
Legal maneuver choices represented by arrows.

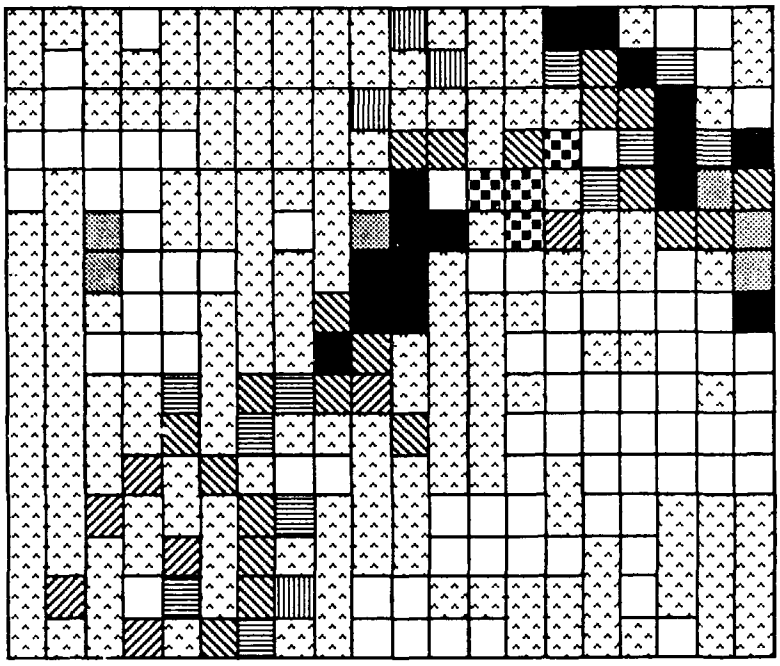


Figure 5c: A reduced terrain map, n=16, m=20.

The final step involves a classification override due to the high probability of threat. If the briefed threat exposure probability of a given region exceeds threshold, its individual coordinates are examined to determine the cumulative probability detection for the region. If sufficiently high, the entire region is designated as threat and labeled appropriately. Figures 5b and 5c indicate the classification of each region in the checkerboard by a shaded intensity legend. Figure 5a shows the map which was processed to produce Figures 5b and 5c. The internal representation of the the checkerboards corresponds to an n x m array where numbers are assigned to the 9 region classifications. States in the representation consist of these arrays together with positions of the opponents on the map. The state transitions will be represented by pairing to each state a maneuver choice 1 through 8 which represent maneuvers to one of the neighboring squares in the checkerboard (see Figure 5b).

Several mission specific observations reduce the task of preprocessing the maps. First, we are concentrating on the mutually aggressive, lost line of sight fight. So the opponents will be separated by a hill (reducing the number of patterns to process) and be pursuing one another limiting the number of maneuver choices. The expert knowledge is encoded as a maneuver choice, recorded for each state, yielding a desired heading (to one of eight neighboring squares), desired altitude, and desired velocity. The default setting for desired altitude is terrain following. This is currently being used to simplify the problem.

With this specification, the task of accessing the expert knowledge in a timely fashion, reduces to retrieving the pertinent

maneuver choice from the data structure. Whether this is workable depends on the speed of this retrieval and the accuracy of the expert knowledge. Once the maneuvers are found, other knowledge may be brought in to second guess or refine the maneuvers if time is available. This is accomplished by the other submodules of the LLOS planner and can use other parts of the representation hierarchy. Also metalevel constraints which use the detailed DMA maps may be required to adjust the chosen maneuvers for ground avoidance. For each checkerboard grid (numerical array) there is a set of copies of that grid with various positions of red and blue (red is the opponent and blue is ownship), where red can be positioned by a blob (set of adjacent squares), and blue by one square. The blobs are determined, if the maneuver choice will not change for red inside the blob. If red can be in any one of several blobs some conflict resolution strategy must be employed, for instance: choose a maneuver for one of the possible (most likely) red positions. Identifying maneuvers for given map patterns and placement of opponents in the grid is done by "expert knowledge" (see Section 5). Filling in for incomplete knowledge, not making the pilot consider every map is accomplished by assuming like checkerboard patterns yield like responses. Research on efficient storing and retrieving of patterns is described in Section 7.

If maps must be processed at every level, and at every resolution, these techniques are too cumbersome. However, for each mission type and general terrain description, there is a minimum sector size and resolution which is the pertinent one to preprocess. First of all, if the opponents are far apart, then there is time to use some symbolic (middle level) reasoning for maneuver choices. The QUICK RESPONSE module for which this representation is designed is for opponents at fairly close range, probably separated by one nill formation. Therefore maps which will be pre-processed for the top level representation will be of much smaller area than the example depicted in Figure 5a. As usual one will choose the lowest level in the quadtree such that both opponents are contained in the grid.

4.2 MIDDLE LEVEL

The middle layer of the hierarchy will consist of terrain descriptions in terms of tactical terrain primitives providing a symbolic description of the lay of the land. Work on this level of the representation is at an early stage. The following reports some preliminary ideas. At this level reasoning about the terrain can be accomplished by inference rules. McDermott and Gelsey (1987) list four military aspects of terrain 1)observation and fire, 2) concealment and cover, 3) obstacles and 4) avenues. Methods for this level resemble previous work [McDermott and Gelsey 1987; Kuan 1984] except that the maps are partially preprocessed, and the primitives are different for the air battle. We propose that an initial list of tactical aspects of terrain for the air battle might include: 1) avenues or corridors, 2) obstacles (either obstructing the desired flight path or

obstructing the desired sensor line-of-sight), 3) hiding places including momentary hiding for purposes of conf sing opponents or weapon systems, 4)areas for reducing the capacity of sensors (high clutter regions, optical camouflage), 5) traps (dead-ends for potentially trapping or otherwise forcing the maneuvers of the opponent), 6) potential emergency landing sites. The terrain maps cannot be completely preprocessed for reasoning at this level since tactical aspects are too context dependent. However, some pre-processing in terms of the tactical terrain primitives listed above can attach labels to regions. Multiple labels may be attached with the pertinent one being chosen using contextual information at execution time.

Actual tactical use of terrain is mission dependent. Fast, non-maneuvering vehicles (bombers) may not consider terrain for anything other than avoidance and following. Slower, maneuverable vehicles (helicopters, CAS aircraft) may take maximum advantage of terrain. In some instances, either opponent may choose to simply increase altitude to avoid fighting in the terrain. In viewing terrain as obstacles obscuring view of the opponent, high level symbolic reasoning about terrain may generalize to other factors which obscure view, i.e. clouds, smoke, position of the aircraft under or behind ownship, lost view by virtue of pilot overload, etc. It is common to treat certain non-moving threats such as stationary SAM sites, ground early warning radars, etc., as obstacles which can be treated similarly to terrain by some algorithms.

4.3 LOW LEVEL

The lowest layer consists of the unreduced world representation, DMA map sections, aircraft positions, velocities, etc., with procedural knowledge contained in traditional (numerical) models. It is possible within REACT to use inference rules at this level to manipulate models using heuristics [Hayslip and Kirsch (1988)]. It is also possible that at a later stage some on-line processing of maps directly from the DMA data may be necessary (see Section 7).

5. KNOWLEDGE ACQUISTION PHASE AND PATTERN GENERALIZATION

The terrain maps must be reduced (to checkerboards) and a direct interactive method developed whereby expert tactical decisions can be acquired for given states . These can then be used directly for short term implementation and recorded so that patterns between states and responses might be recognized and aggregated at a later stage of the project. Knowledge acquisition for the high level of the representation hierarchy amounts to showing a pilot a (3-d rendering of) terrain in the immediate area, from the viewpoint of his aircraft, telling him the general expected vicinity of the opponent (if known) and asking for his maneuver choice. This response would then be immediately entered as a maneuver choice, coupled with the corresponding reduced terrain pattern (array of numbers) in the internal memory. In other words, the pilot sees the "real" terrain but his response

is immediately stored in the high level knowledge structure.

Any knowledge base is an incomplete and approximate model. It is impossible for a pilot to pre-process all map segments. Therefore, storing terrain patterns and maneuver responses for large, complex map areas is not feasible, so this method is only used for the reactive fast response module of REACT. Even for the relatively short range cases, there are may possibl_ terrain patterns. If the terrain is rough, it may produce too many fine patterns for this processing. In that case the terrain is smoothed, and the fine details are ignored. This corresponds to the system ignoring those details for lack of processing time. A number of features in the patterns have emerged, repeating pattern types for the close in cases. Given these reductions in the number of patterns, pilot responses can be monitored, and then maps can be preprocessed based on generalizations over pilot responses to similar situations (see Section 7).

During knowledge acquisition for the high level knowledge representation using the interactive tool and the dual map representations , some symbolic information which can be encoded in rules for the middle level representation may be acquired. Answering the "why" after prototypical responses may yield knowledge which can be encoded in rules. Other top level information which should emerge includes 1) reasonable minimum cell size (level in quadtrees) for preprocessing, 2) minimum necessary resolution for the checkerboards to provide adequate information, 3) bounds on the quantization error associated with choosing only maneuvers as checkerboard moves, 4) better operational definitions of tactical primitives for the middle level representation, 5) pattern types for the high level representation which can be stored for generalization. Also metalevel rules can be extracted, i.e. when to ignore the terrain altogether, when to slow down in order to provide more time to "think", etc.

6. IMPLEMENTATION STATUS

The REACT prototype system is being developed on the Symbolics 3675 computer using the Georgia Tech Generic Expert System Tool (GEST) with numerical procedures (including AUTOMAN and procedures from TOPCAT) being implemented in FORTRAN and C on a Data General MV/10000. Due to the complexity and scope of the REACT system, implementation efforts have been concentrated on incremental developments designed for maximum flexibility and extendability of the system as a research vehicle. This includes provision for simulation where the architecture design calls for concurrent processes. The groundwork and initial stages for coupling the experimental planner were implemented this year including the integration of AUTOMAN and terrain maps. Integrating REACT, the scoring function and game theoretic approach of AUTOMAN and heuristics in a functionally coupled system also allows AUTOMAN guidance law routines tc be used in the plan executor for the LLOS planner, and it will allow multiple aircraft scenarios to be developed. A third advantage afforded by the integration of

AUTOMAN into REACT is the potential for "flying" the LLOS experimental planner against an AUTOMAN opponent with perfect information, resulting in a test arena for the planner. Initial implementation work on processing the terrain maps for the high level representation was completed by Georgia Tech Research Institute.

7. CONCLUSIONS AND EXTENSIONS

Eventually the pairings of maneuver choices to battle states in the top level representation might exhibit patterns which can be learned by the system. In that case maneuvers could be chosen or adjusted on the fly rather than being prestored in the representation. This greatly reduces the amount of map analysis which must be done before the flight. The discrete representation in terms of grid arrays with numbered maneuver responses, means that the pairings of patterns to responses look like the pairings of input to output vectors that are being considered for neural network applications. There is inefficiency (lack of conceptual economy) arising from representing the same knowledge repetitively in the structure (map sectors which look alike). If a network could be utilized to match patterns to responses, then the duplication in the database is reduced. Eventually a concurrent system might look ahead, retrieving maneuver choices for adjacent squares, and further if time allows.

Any abstraction of the world into an internal representation is an approximation and therefore it will be necessary to analyze potential errors resulting from this reduction. We mention some potential errors and inefficiencies which could arise from the reduction of the world into our representation. In particular the "quantization" errors arising from choices in the resolution of DMA map processing , and the limiting of maneuver choices to be only toward adjacent squares in a grid must be analyzed. Some problems may arise when one tries to tune the representation for other non-terrain factors such as introducing obstacles/hills for pop-up SAM sites, or changing the maneuver quantization thresholds to account for aircraft turn radius restrictions. Also, maintaining consistency in the database will become a problem in some cases . For example if meta-knowledge is used to temporarily adjust numbers in the grids, then the database must be restored to the original terrain form in case the aircraft returns to that sector. Examples of metalevel reasoning include the following. If no SAM sites are around, then there is more freedom to disregard terrain, perhaps to fly higher concentrating on using terrain only for masking from the opponent. We have so far avoided the navigation and mapping problem. That is, we assume the aircraft knows where it is over the map. However, in the design of the top level representation, the aircraft only needs to know what square in the checkerboard it is in. So in this sense, the utility of the high level representation is not as sensitive to navigational error. Continued research on these and related issues, with efforts to map the architectures onto the appropriate parallel hardware is necessary for incrementally pushing coupled AI systems toward realtime

military application capability. In particular, AI researchers must consider features of realtime conventional software development and realtime database management.

REFERENCES

Austin, F., Carbone, G., Falco, M., Hinz, H. 1987: *Automated Maneuvering Decision for Air-to-Air Combat,* Proceedings AIAA Guidance, Navigation and Control Conference, Monterey, California.

Brooks, R.A.1986: *A Robust Layered Control System for a Mobile Robot,* IEEE Jour. of Robotics and Automation, RA-2, No. 1.

Falco, M., Carpenter, G., Kaercher, A.1974: *The Analysis of Tactics and System Capability in Aerial Dogfight Game Models,* Grumman Corporate Research Department Report No. RE474, May, 1974.

Georgeff M., and Lansky, A. 1987: *Reactive Reasoning and Planning,* Proceedings AAAI 87.

Gilmore J. and Semeco, A. 1986: Knowledge-based Approach Toward Developing an Autonomous Helicopter System, Optical Engineering, 25(3), 415-427.

Hayslip I. and Kirsch R. 1988: *On Focusing and Adaptive Planning,* Proceedings Applications of Artificial Intelligence VI, S.P.I.E.

Hayslip, I. and Rosenking J. 1988: Adaptive Planning for Threat Response, (submitted S.P.I.E. Applications of Artificial Intelligence VII) Grumman Corporate Research Center, Bethpage, N.Y.

Kaelbling, L.P. 1986: *An Architecture for Intelligent Reactive Systems,* Proceedings of Reasoning about Action and Plans Workshop, Timberline Ore., M. Georgeff and A. Lansky, ed., Morgan Kaufman.

Kuan, D.1984: *Terrain Map Knowledge Representation for Spatial Planning,* Proceedings, First Conference on Artificial Intelligence Applications.

McDermott, D. and Gelsey, A.1987: *Terrain Analysis for Tactical Situation Assessment,* Proceedings 1987 Workshop Spatial Reasoning and Multi-Sensor Fusion, Morgan Kaufmann,Los Altos, CA.

Rosenking, J., and Roth, S. 1988: *REACT: Cooperating Expert Systems Via a Blackboard Architecture,* Proceedings SPIE Applications of Artificial Intelligence VI, 1988.

Roth, S., Tynor,S., Gilmore, J., Mendelsohn, J.1987: *GEST-Development of a Blackboard Expert System Tool,* Workshop on Blackboard Systems Implementation Issues, AAAI.

Samet, H.1981: *An Algorithm for converting rasters to Quadtrees,* IEEE Trans. on Pattern Analysis and Machine Intelligence, 3.

FUTURE MINEFIELD TERRAIN ANALYSIS REQUIREMENTS

Robert A. Sickler
U.S. Army Engineer School
Directorate of Combat Developments
Ft. Leonard Wood, Mo. 65473-5000

ABSTRACT

Since early in its employment the land mine's principle role has been that of an obstacle. Unfortunately this classical application does not adequately meet the anticipated mine warfare needs of Airland Battle Future. Minefields of the future must not only be directional obstacles but they must also be 'stand-alone' lethal weapon systems, capable of both offense and defense action. The concepts and capabilities inherent with compliance of these needs leap far ahead of conventional systems. Minefields of the future (Intelligent Minefields) will require highly lethal munitions, sophisticated sensors, and expert system control. The operational complexity of technology associated with these systems will generate the need for extensive engineering in both minefield site selection and in designing the minefield to fit the terrain. As a result engineers responsible for building future minefield will be overcome by the sheer number of parameters that must be analyzed. It is therefor evident that if we are to capitalize on the potential of the Intelligent Minefield we must automate the analysis of tactical terrain data for minefield design.

1. INTELLIGENT MINE FIELDS

It is more accurate to say 'Intelligent Minefields in conception' rather than Intelligent Minefields because the inventory of existing mines and their utilization is little changed from that of World War II or the Korean War. Modern mine systems are needed however, and Intelligent Minefields (IMF) could serve as a key element of the countermobility portion of the Engineering and Mine Warfare Mission for Air Land Battle Future.

The objective of IMF in the future would be to provide a means of both countering threat mobility and at the same time degrading threat forces. To achieve this goal Intelligent Minefields (IMF) will need to possess a stand-alone

capability, that is to say that they must function without manned observation/control and without covering fire. This capability will enable the IMF to serve as force multipliers in gaps or lightly manned sectors, on a battlefield where friendly forces are at a minimum. The role of the IMF then is to increase threat force attrition through increased obstacle lethality, to degrade threat mobility through control of individual mines, and to counter threat breaching efforts through the use of selective targeting mines. The only way to achieve this level of operation is to have designed the minefield as an integral part of existing terrain.

2. TERRAIN INFLUENCE ON OPERATION

As initial NATO fronts begin to roll back under the weight of opposing forces and as Soviet Operational Maneuver Groups (OMG) move deep into NATO territory the battlefield will take on a fluid nature. At this time maneuver commanders will be required to optimize force deployment, to meet key threat advances, and still remain in tactical deployment. As a result, the commander will be faced with an ever changing array of weak areas and gaps in his defence structure.

To keep his forces at a functional density and still cover a reasonable front the friendly battlefield commander could deploy Intelligent Minefields along key terrain features. This would allow manpower and firepower to be concentrated at other portions of the contested area while the minefield is left, at least temporarily, to hold the gap or weak area. In this mode the IMF will deviate from the conventional minefield by functioning as an unmanned flexible barrier which possess both Terrain Obstacles and Lethal Weapon Systems.

The typical IMF would be configured with two parallel mine belts that are separated by a Wide Area Mine Killing Zone, Figure 1. The belts will be composed of conventional mines and improved conventional mines that have been located along terrain features that will accentuate both their concealment and their role as an obstacle. The outer belt, first to be encountered by the threat, will consist primarily of mines that have been buried, to help them escape detection during threat reconnaissance. The inner belt could be designed to represent a 'hasty' conventional minefield or it could be an mine belt similar to the outer belt. The area between the mine belts will be covered by smart mines that cover a wide area of the minefield. These mines will utilize terrain features to maximize there effectiveness in targeting vehicles and to reduce the possibility of detection. Of paramount importance is the realization that the IMF is not composed of rigid geometrically configured blocks of mines but that it will be a complex series of mine obstacles woven into the terrain fabric.

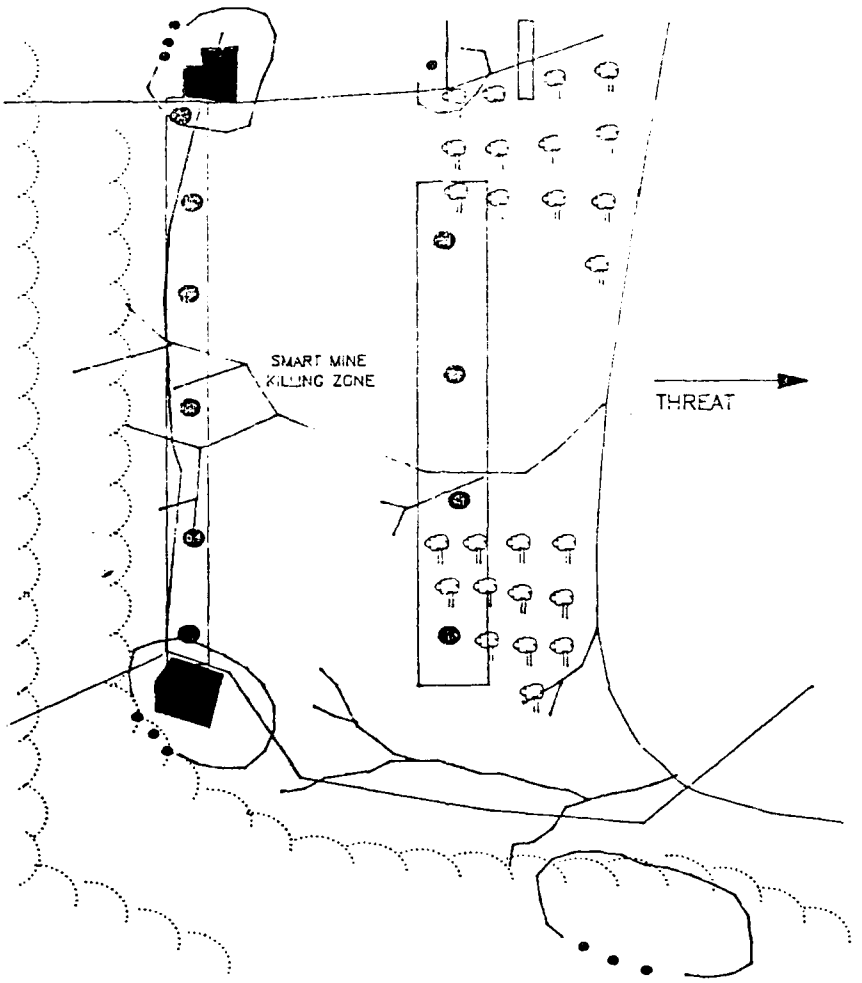


Figure 1. A typical Intelligent Minefield constructed to hold key terrain between two villages.

The entire IMF will remain in a latent mode, functioning as an integral part of the terrain, during early probing by threat recon patrols. In this manner the area being probed by threat recon will appear to be a weak area or a gap in friendly lines. The threat commander, being locked into a rather rigid set of lower unit level tactics, will be forced into trying to take advantage of what he perceive to be a gap in enemy lines. Through these tactics those portions of the defense manned by the human element will be relieved of pressure as the IMF is engaged by the brunt of threat forces.

As threat forward security elements or perhaps advance guard lead elements pass the outer mine belt, the IMF main computer goes to alert status. Threat forces will proceed through the central killing zone unimpaired and just as lead elements enter the inner mine belt, the IMF main computer activates mines in this belt. This inner belt will stop forward movement of the lead element causing following forces to either pile up in the Smart Mine Killing Zone or come to a complete stop. As the density of threat vehicles reaches a threshold level or threat vehicles reverse their direction of travel the outer minefield is turned on. This then traps threat vehicles between the two minefield belts where Wide Area Mines will begin a systematic engagement of threat systems.

3. TERRAIN ANALYSIS FOR SENSOR LOCATING

There are three basic sensor types that will be employed in the design of the Intelligent Minefield. The location of these sensors will either be outside the minefield in an 'over-watch' role or within the minefield to enhance sensor resolution. The principle 'over-watch' sensor will monitor energy in the visual region and therefor must be located where there is a substantial view of the battlefield. Visual sensors offer both the range and passive capability to allow them to watch for early signs of threat activity without being detected themselves. Those sensors located within the IMF will be either acoustic or seismic energy sensors. Because the medium of energy transmission is not as sensitize to blockage as visual these sensor can be more readily concealed within the minefield.

Terrain analysis for visual sensor locations will be predicated on the concept of minimizing areas of innervisibility. That is to say that the inferencing mechanism must strive to find a location where the sensor can see and yet not be seen. The key two features that must be evaluated therefore, are land forms and vegetation. A chosen location must offer; the proper relief for observation, enough vegetation for concealment, yet not so much vegetation that it obscures the view.

The placement of acoustic sensors will be a function of vegetation and relief but in a different manner than the visual sensors. A key parameter in determining the location of acoustic sensors will be to insure that the energy signal has not become distorted nor has not lost its directional attributes. This will be a problem in areas where the relief is so great that acoustic energy waves become reflected and refracted, to the point where they no longer posses their original characteristics. Although vegetation may contribute to the loss of wave characteristics it's main degradation of the signal will be through attenuation of signal strength.

Determining proper locations for seismic sensor requires a more detailed analysis of terrain features than for any of the other sensors. Terrain attributes will not only affect the receiving of seismic energy but they will also have a tremendous impact on the generation and transmission of seismic energy. In an area where the soil is hard and rocky, vehicles will generate more seismic energy and this energy will be readily transmitted to the substructure. Conversely, areas where the soil is soft and sandy will serve to reduce seismic energy generation and to dampen seismic energy transmission to the substructure. Landform and substructure will also influence the rate of seismic energy attenuation between the energy source and the sensor. At the sensor itself the soil will play a large role in how well and how much seismic energy is conveyed from the substructure to the sensor. The proper locationing of seismic sensor must take all of these terrain features into account as well as those parameters associated with providing cover and concealment for the sensor itself.

4. THE NEED FOR RESEARCH

For the IMF to function at its fullest, it must rely on technology that is just now emerging from the fields of computer science, vision, communications, sensing, and robotics. Deployment and component complexity of this technology will vary with the amount of 'stand-alone' capability, range of coverage, and the degree of lethality required of the minefield. Although there is a great diversity in IMF, all will contain three basic systems; mines, sensors and expert systems. The more complex and larger IMF will also contain ROBOTIC vehicles used in preparatory mine emplacement and situational mine emplacement.

At present there are a number of institutions conducting research relative to the technologies that will be an integral part of the IMF, however there is very little research being conducted in the areas of minefield design. There exists a strong need for research aimed at determining a relationship between IMF components and the terrain they will be defending. This work is needed to insure that terrain attributes accents the operation of the IMF rather than to degrade its operation.

MERCURY: A MESOSCALE METEOROLOGICAL DATA FUSION SYSTEM FOR CORPS-LEVEL APPLICATION[1]

C. A. Fields, M. J. Coombs, C. Cavendish, T. C. Eskridge, R. T. Hartley,
H. D. Pfeiffer, and C. A. Soderlund
New Mexico State University
Las Cruces, NM 88003-0001, USA

S. Kirby and G. McWilliams
U. S. Army Atmospheric Sciences Laboratory
White Sands Missile Range, NM 88002-5501, USA

ABSTRACT

The MERCURY mesoscale meterological data fusion system will be an intelligent, autonomous interface between the Integrated Meteorological System (IMETS) meteorological database, the Digital Terrain Support System (DTSS) terrain database, and Tactical Decision Aids (TDAs) that require meteorological data that has been fused with terrain and land use data. This paper describes the MERCURY task environment, and the architecture being developed to meet the demands that it imposes.

1. OVERVIEW OF THE MERCURY PROJECT

1.1 THE ROLE OF MERCURY IN IMETS

The MERCURY system is being developed as a component of the U.S. Army Integrated Meteorological System (IMETS), a software system for Corps level acquisition, management, integration, and distribution of meteorological data, nowcasts, and forecasts. MERCURY will be responsible for data integration within IMETS. IMETS will provide meteorological information to the All Source Analysis System (ASAS), the Army's Corps and division level intelligence analysis system; MERCURY will, therefore, perform meteorological data integration within the ASAS context. MERCURY has been under development since August, 1987. The requirements to be met by MERCURY and the technologies available for MERCURY implementation are reviewed in Coombs et al. (1988).

[1] This work was supported by U. S. Army Atmospheric Sciences Laboratory Contract DAAD07-86-C-0034 to New Mexico State University. The views, opinions, and findings contained in this report are those of the authors, and should not be construed as an official Department of the Army position, policy, or decision unless so designated by other documentation.

The role of MERCURY within IMETS is illustrated in Fig. 1. MERCURY will serve as an autonomous, intelligent interface between the IMETS meteorological database and tactical decision aids (TDAs) that require meteorological data as input. The task of MERCURY is to provide each TDA with appropriate meteorological data for the locations and times for which the TDA is run. MERCURY is intended for use in a mesoscale domain corresponding roughly to the area of operation of a Corps, i.e. a domain of 300 km × 300 km or less. MERCURY will primarily operate as a nowcasting tool, i.e. in a temporal range from current to approximately six hours in the future. The TDAs served by MERCURY will include, for example, programs for visibility and transmissivity assessment, smoke screen production, nuclear-biological-chemical (NBC) airborne particulate and aerosol dispersion, surface trafficability, and low-level wind prediction for flight operations. Most current TDA programs require numerical values of meteorological variables such as temperature or wind velocity as input; however, future TDAs may also require qualitative descriptions of weather features or patterns, such as the presence of a front boundary, as input.

MERCURY will have access to current and recent data obtained by meteorological data sources, including mesonet ground stations and rawinsonde stations, within the 300 km × 300 km area of interest. Data from additional passive instruments, such as thermal and wind profilers, instruments flown on remotely piloted vehicles (RPVs), and satellite instruments, may be available in the future. These data will be stored, after quality assessment and error rejection or correction, in the IMETS database. The IMETS database will also contain derived numerical products provided by the Air Weather Service, including synoptic objective analysis or other diagnostic model output, and prognostic model output, for an area containing the area of interest. Climatological data for the area of interest will generally also be available. Both the derived products and the climatological data will typically be of relatively low resolution in space, time, or both; the live data will have variable sampling density in both space and time.

Meteorological data for the exact location and time required for most TDA applications typically will not exist in the IMETS database. The locations for which data are required may, for example, be in enemy hands, and may be separated from the locations for which data are available by significant terrain features or by a considerable distance. In such cases, MERCURY will be required to generate the appropriate data from the data that are available. This may require extrapolation in space, time, or both from the data that are available in the IMETS database, combination ("fusion") of data from different sources and of different resolutions, and integration of available meteorological data with terrain and land use data. These operations must be performed transparently to the TDA, i.e. the generation of extrapolated data must be indistinguishable, from the perspective of the TDA, from the direct retrieval of data from the database. MERCURY must, therefore, operate in near real time. MERCURY will, moreover, typically communicate directly only with the relevant TDAs and with the IMETS database; no user interaction with MERCURY is anticipated during normal operation in the field.

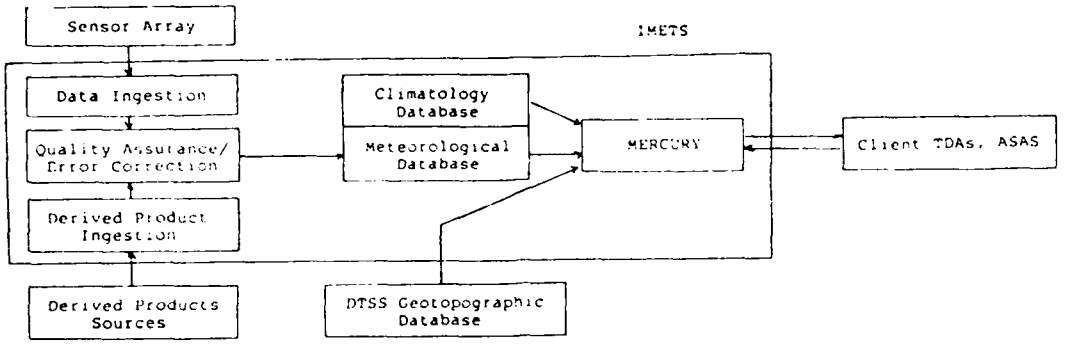


Figure 1. Block diagram showing information flow in the IMETS system. MERCURY will serve as an intelligent interface between the IMETS database and the TDAs that require data from IMETS. MERCURY will access terrain and land use data directly from the Digital Terrain Support System (DTSS) geotopographic database. From the perspective of MERCURY, DTSS may be considered a database, while ASAS may be considered a TDA.

1.2 THE NEED FOR A NOVEL AI APPROACH

The interpolation or extrapolation of meteorological data to produce gridded fields is currently carried out using objective analysis. Available objective analysis schemes range from simple weighted-averaging procedures to considerably more complex procedures that utilize terrain following coordinates and enforce constraints based on the physical dynamics of the atmosphere. All objective analysis schemes suffer, however, from two drawbacks: they require synoptic data sets, i.e. data sets obtained at a single time, and they are unreliable for extrapolation beyond the spatial limits of the data. Given that significant extrapolation will often be required of MERCURY, and that MERCURY will often have to use data obtained at different times, objective analysis alone is insufficient in the MERCURY task environment.

The fusion of data from different sources, e.g. from ground stations and satellite images, and the fusion of data with synoptic objective analyses or numerical

predictions, is currently carried out interactively by expert meteorologists using fax charts or workstations such as the Satellite Data Handling System (SDHS) or the Portable ASAS Work Station (PAWS). This process is time consuming, and requires considerable expertise. It is impractical, if not impossible, under the time stress of a battlefield environment in which many TDAs need different types of data for different locations almost simultaneosly. Interactive data fusion, moreover, requires highly skilled personnel.

The above difficulties with current objective analysis and interactive data fusion techniques provide the motivation for the MERCURY project. To succeed in the MERCURY task environment, a software system must combine the meteorologist's ability to fuse data from different sources and to extrapolate the results to other locations with the speed and autonomy of objective analysis and numerical prediction. An artificial intelligence (AI) approach has been taken to meet this goal.

Expert system technology has thus far been the most widely applied AI technology. A number of existing meteorological expert systems have been reviewed by Dyer (1987). Most existing meteorological expert systems have been developed to generate qualitative predictions of particular weather patterns for particular locations or types of locations, e.g. upslope snow storms in the lee of mountain ranges. These systems rely heavily on location-specific heuristics. The use of such heuristics is impractical for a system such as MERCURY, which must be useable in any geographic location. The use of a simulated neural network, such as that of Young (1987), is similarly impractical, because such networks must be trained using extensive historical data for the locations at which they will be used.

In the Final Report for the recommendations stage of the MERCURY project, Coombs et al. (1988) concluded that no existing AI technologies for automated problem solving, including second generation expert systems technology, were sufficiently flexible for direct use in MERCURY. The development of a new automated problem solving technology - model generative reasoning (MGR) - specifically designed for qualitative data fusion tasks was recommended. The development of this technology has proceeded in parallel with the implementation of the first MERCURY prototype (Coombs and Hartley, 1987, 1988; Fields et al., 1988).

2. THE **MERCURY-1** PROTOTYPE

2.1 OVERVIEW

An initial MERCURY prototype, MERCURY-1, was developed in order to gain familiarity with the data that would be employed by MERCURY and the functionality requirements of the MERCURY task environment. The MERCURY-1 effort focussed on developing prototypes of the geotopographic representation and the developer interface, as the structure and functionality of these components needed to be established before further development could proceed. MERCURY-1 employs a straightforward point representation for mesonet and rawinsonde data; it does not

represent gridded data. The data analysis component of MERCURY-1 employs heuristics encoded as simple numerical functions to extrapolate meteorological data to locations for which data are not available. Design of the MERCURY-1 prototype began in August, 1987, and implementation of MERCURY-1 on the Symbolics 3650 under Genera 7.1 began in September, 1987. Development work on MERCURY-1 ended in June, 1988.

An approximately 300 km × 300 km region surrounding the Los Angeles, California basin was selected as an initial test location for MERCURY prototypes. This region includes oceanic coastline, the Los Angeles urban area, the San Gabriel mountains, and the Mojave desert. The region exhibits a number of meteorological phenomena of interest, including sea and mountain breezes, and frequent passage of both inland and off-shore high and low pressure systems. A dense set of both surface and upper-air data has been obtained for this region by ASL, and is available for testing MERCURY prototypes.

2.2 ARCHITECTURE

The MERCURY-1 architecture is shown in Fig. 2. The architecture reflects an early design decision to separate the data analysis functions of MERCURY into two components: a component utilizing conventional numerical or AI techniques, and a component utilizing the MGR system being developed in parallel with MERCURY. In MERCURY-1, the former component implements a set of heuristics for evaluating the likelihood that data obtained from a particular source are representative of an unknown location; hence it is termed the data evaluation module (DEM). In the MERCURY-1 design, it was assumed that the DEM would be supplemented, at some point in the future, by an MGR-based scenario generation module (SGM) that would generate qualitative descriptions of meteorological scenarios from the available data. Because the MGR software was still under development, the SGM was not implemented as part of MERCURY-1.

The MERCURY-1 geotopographic representation is based on the commercial GeoFlavors software, which was purchased from Ball Systems Engineering Division (formerly Verac, Inc.), San Diego, CA. The developer interface of MERCURY-1 includes functions for specifying geographical regions of interest and editing maps of these regions. Maps may include land use regions, terrain contours, and locations of mesonet and rawinsonde stations and test locations. The MERCURY-1 developer interface, with a map of the Los Angeles basin test area, is shown in Fig. 3.

A live source of meteorological data was not available during the period in which MERCURY-1 was under development. Data input to MERCURY-1 is, therefore, performed by the developer via a set of developer interface functions. The developer also simulates the client TDA by requesting data for a particular location and time.

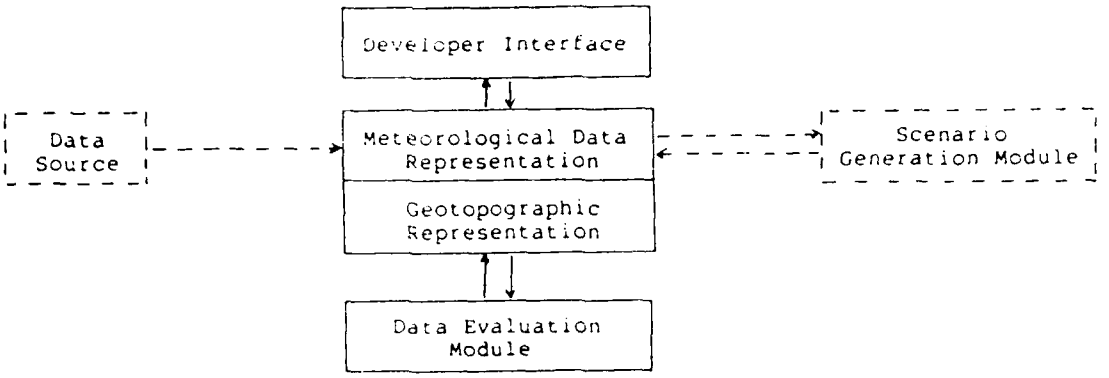


Figure 2. Architecture of the MERCURY-1 prototype. The components indicated as dashed boxes were not implemented in MERCURY-1; their roles in the architecture are included for clarity.

2.3 DATA EVALUATION MODULE

The MERCURY-1 data evaluation module implements a set of heuristics for choosing, from among those available in the region of interest, a single mesonet station and a single rawinsonde station as most likely to provide data representative of the current weather conditions at the location of interest. Qualitatively, these heuristics are as follows: 1) nearby stations are more likely to provide representative data than far away stations; 2) stations reporting recent measurements are more likely to be representative than stations reporting old measurements; 3) stations located in similar land use regions are more likely to provide representative data than stations located in dissimilar land use regions; 4) stations separated from the location of interest by significant terrain features are less likely to provide representative data than stations that are not separated from the location of interest by significant terrain features. All of these heuristics are used in evaluating mesonet stations for representativeness; only heuristics 1) and 2) are used for evaluating rawinsonde stations.

The above heuristics are encoded as numerical functions that evaluate all available measurement stations with respect to the location of interest. Numerical functions were used for two reasons: first, they provide a straightforward way of handling the numerical input data, and second, the results of different heuristic evaluations can be combined as continuous values. Stations are evaluated using a quality function that additively combines component functions corresponding to the four heuristics. This quality function measures the likelihood that a station will provide the optimal data for the location of interest. Quality is assumed to decrease exponentially with distance, age of the data, and difference in land use as represented by roughness parameter, and to decrease inversely with the difference in

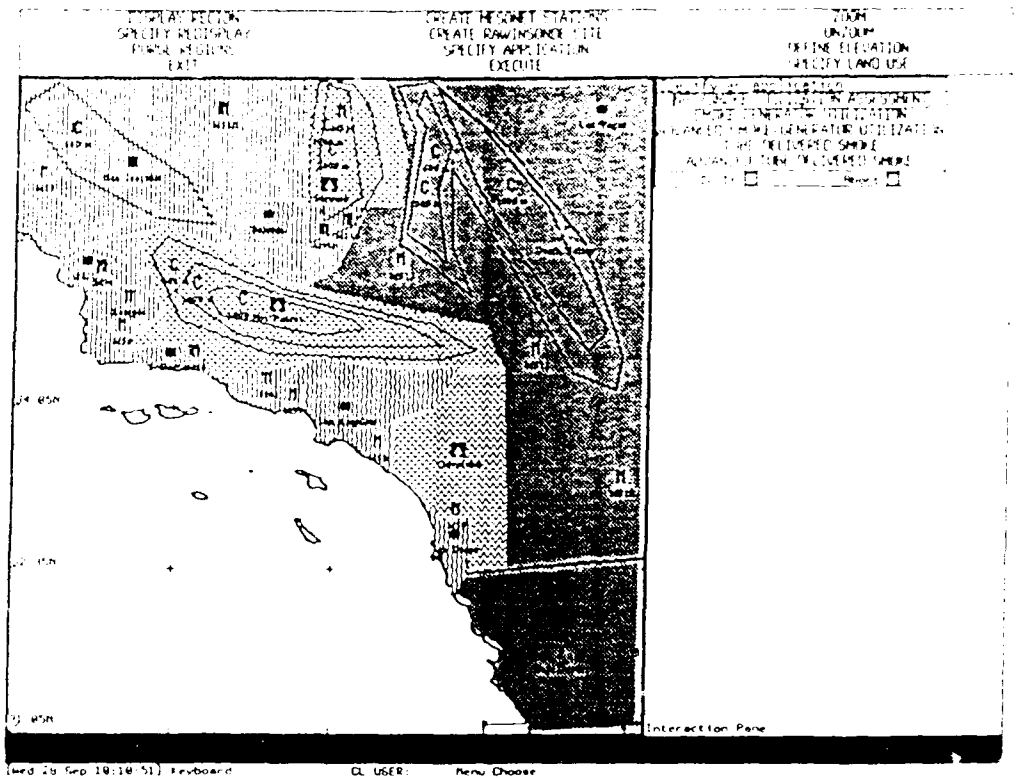


Figure 3. Los Angeles basin region, showing shading patterns for land use regions. The interaction window shows the application selection menu.

elevation. The quality, $Q(m)_j$, of the j^{th} mesonet station with respect to a particular location of interest l is given by:

$$Q(m)_j(l) = \Sigma_{i=1-3}\, \alpha_i \, \exp\,(x_{ij}(l)/x_{0i}) + \alpha_4 \,/\, \int_{l\rightarrow j} |\Delta\,(x_4)|,$$

and the quality, $Q(r)_k$, of the k^{th} rawinsonde station with respect to l is given by:

$$Q(r)_k(l) = \Sigma_{i=1-2}\, \beta_i \, \exp\,(x_{ik}(l)/x'_{0i}),$$

In these expressions,

$x_{1j}(l)$ = distance from j^{th} station to l.

$x_{2j}(l)$ = difference in time between the last report from the j^{th} station and the present.

$x_{3j}(l)$ = difference in roughness between the location of the j^{th} station and l.

x_4 = elevation above mean sea level. In MERCURY-1, the path integral is approximated as the sum of the contour boundaries crossed by a straight line from l to the location of the j^{th} station.

α_1, α_2, α_3, α_4, β_1, and β_2 are positive constant coefficients, and x_{01}, x_{02}, x_{03}, x'_{01}, and x'_{02} are positive constant 1/e lengths.

These expressions contain a total of six coefficients and five 1/e lengths, for a total of eleven free parameters.

2.4 EVALUATION OF MERCURY-1.

The goal of the development of MERCURY-1 was to investigate the requirements posed by the MERCURY task environment, and to serve as a testbed in which to evaluate architectural approaches to meeting the requirements posed by the task environment. The development of MERCURY-1 has provided answers to a number of these questions. The principal conclusions reached on the basis of MERCURY-1 were that a topographic data representation based on point elevation and land use data, a live source of meteorological data, and a significantly expanded data evaluation system were needed (Fields, 1988). These conclusions served as the basis for the MERCURY-2 design.

3. THE MERCURY-2 SYSTEM

3.1 OVERVIEW

The goal of the development of the MERCURY-2 prototype is to provide a testbed for evaluating approximate meteorological diagnostic techniques and objective analysis strategies in a data fusion environment, and for developing specifications for the MGR based qualitative modelling system. The design of MERCURY-2 is similar to that of MERCURY-1, but incorporates the alterations recommended in the MERCURY-1 evaluation. Foremost among these are facilities for ingesting live meteorological data, and a geotopographic representation based on high-resolution Defense Mapping Agency (DMA) gridded elevation and land use data. This geotopographic representation will interface directly with a meteorological data representation based on the UniData System for Scientific Data Management (USSDM) package developed by the University Corporation for Atmospheric Research (UCAR) UniData Program (Campbell and Rew, 1988).

The MERCURY-2 design includes a heterogeneous data analysis system that replaces both the Data Evaluation Module and the unimplemented Scenario Generation Module of MERCURY-1. This system incorporates mesoscale objective analysis, approximate diagnostic models that describe particular weather patterns such as sea breezes, heuristic rules, and qualitative models constructed using MGR. The system is designed to use the faster, more reliable objective analysis or

diagnostic models when input data to drive them are available, and to fall back on a combination of heuristics, qualitative modeling and climatology when data are unavailable. Heuristic metarules are used to direct the flow of control to either quantitative or qualitative models, depending on the data that are available and their characteristics.

3.2 ARCHITECTURE

A data flow diagram of the MERCURY-2 architecture is shown in Fig. 4. MERCURY-2 is organized along the same general lines as is MERCURY-1: it includes geotopographic and meteorological data representations that emulate the DTSS and IMETS databases, respectively, and a data analysis system that carries out data fusion. Like MERCURY-1, MERCURY-2 includes a developer interface for set-up and testing. The principal high-level architectural differences between MERCURY-2 and MERCURY-1 are that MERCURY-2 accepts input directly from meteorological data and derived products sources, and accepts requests from and returns responses to client TDAs.

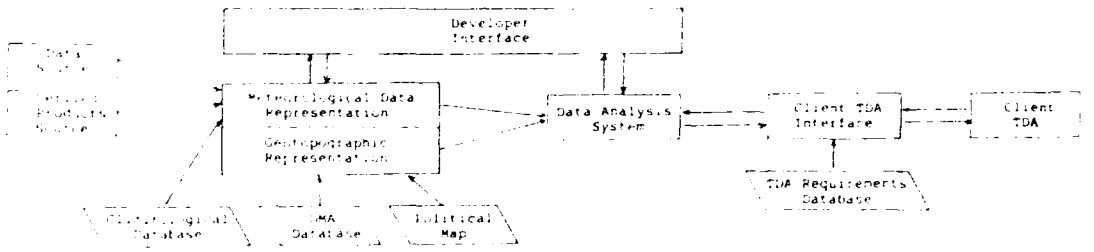


Figure 4. Block data flow diagram of the MERCURY-2 architecture; cf. Fig. 2 for MERCURY-1. Diamonds represent bottom-level databases. The geotopographic and meteorological data representations are shown in more detail in Fig. 5; the data analysis system is shown in more detail in Fig. 6.

MERCURY-2 is designed to accept the same types of meteorological data as did MERCURY-1: mesonet ground station and rawinsonde data. Quality assurance is assumed to be performed on these data by an upstream system (cf. Fig. 1). MERCURY-2 will also accept synoptic objective analyses for an area containing the 300 km × 300 km region of interest; such analyses will typically be relatively low-resolution continental scale products. Objective analyses are assumed to be provided as gridded fields of real numbers representing heights at both mandatory and significant levels, temperatures at 850, 700, and 500 mb, and vorticities at 500 mb (1 bar = 100 kPa). Gridded fields representing numerical predictions will be added as an additional derived input in the second phase of MERCURY-2 development.

Both data and derived products for input to MERCURY-2 will be obtained from the domestic data plus (DD+) and National Meteorological Center (NMC)

numerical products broadcasts of Zephyr Weather Information Service, Inc., of Westborough, MA. These data are corrected for errors by Zephyr. The Local Data Manager (LDM) component of the USSDM software (Campbell and Rew, 1988) will be used to ingest and manage both data and derived products. The LDM will, therefore, emulate the IMETS meteorological database management system within MERCURY-2.

MERCURY-2 will be directly demand driven by its client TDAs. Client TDAs will query MERCURY-2 by passing a request for data for a particular location and time to the client TDA interface, as shown in Fig. 4. The location must be within the current region of interest, and the time must be within a specified interval from the current time. The data requirements of the client TDAs are provided in a TDA requirements database that is accessed by the client TDA interface; a TDA does not, therefore, have to explicitly request values for the variables that it needs. The client TDA interface requests values of the variables required by each TDA that submits a request from the data analysis system, which produces the required values as output. The client TDA interface reformats this output, using formats specified in the TDA requirements database, before passing the result to the requesting TDA as a response to the submitted request.

3.3 DATA REPRESENTATIONS

The organizations of the MERCURY-2 geotopographic and meteorological data representations are shown in Fig. 5. These data representations have been completely redesigned, based on the recommendations made following the MERCURY-1 evaluation. The MERCURY-2 representations effectively form a set of overlays of point data on a bottom-level latitude-longitude coordinate system. All data or derived products associated with a latitude-longitude point may be accessed by the data analysis system via the coordinates of the point. Data may also be accessed by specifying a distance, or a function of distance, from a point. The data analysis system thus treats the geotopographic and meteorological data representations as a single database of point data tied to coordinates.

Point elevation and land use data at 100 m nominal grid resolution (for midlatitudes) are provided by DMA digital terrain databases. In cases in which DMA land use data are unavailable, approximate land use data will be added to the DMA database by hand before using it in MERCURY-2. The DMA database for a region of interest forms the basis of the geotopographic data representation for that region. DMA land use codes are converted to roughness estimates using average roughness values for desert, forest, agricultural, urban, and marine areas (Hansen, 1984). The elevation data are contoured using the National Center for Atmospheric Research (NCAR) Graphics contouring routine, which is included as a component of the USSDM software package. Boundaries of land use regions are represented as lists of coordinates at which the land use code changes. The local slope of the terrain at each grid point is calculated as the cross product of the vectors representing the changes in elevation from the grid point to its nearest neighbors in positive latitude and longitude. Regions bounded by slopes above a specified cutoff are identified as

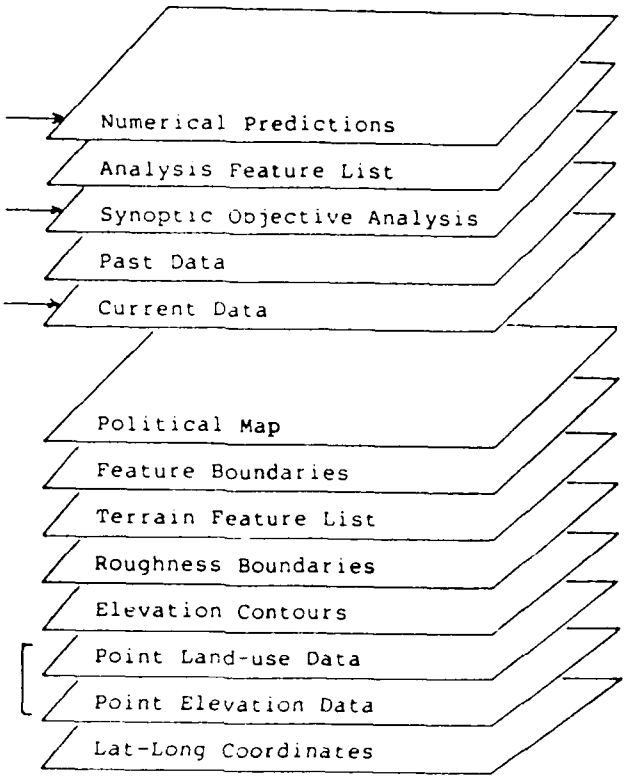


Figure 5. *Organization of the meteorological data and geotopographic representations.* Each overlay is tied directly to the latitude-longitude coordinate system provided by the DMA database. Incoming arrows indicate input from the data and derived products sources; the DMA database is indicated by a square bracket. The other overlays are calculated from those below them as described in the text.

significant terrain features, and are placed in a terrain feature list. Such regions are represented by their boundaries at full resolution, and by the slopes associated with the points inside the boundaries, which are averaged on a 500 m nominal grid (Cavendish et al., 1988).

Also placed in the terrain feature list are the boundaries, again at full resolution, of marine or desert regions above a specified size. The terrain feature list thus provides a compact representation of all features of the terrain that are likely to have significant effects on the mesoscale weather pattern.

Meteorological data are tied to the coordinate system as time-stamped point data associated with the location at which they were measured. Facilities for doing this are provided in the USSDM package. The elevation and land use (roughness) of the location of a measurement station are obtained directly from the relevant geotopographic representation overlay. Distances from measurement stations to terrain features are calculated as Euclidean straight-line distances to their boundaries using latitude-longitude coordinates. Gridded objective analyses and numerical

predictions are represented as additional time-stamped point data overlays on the coordinate system, again using facilities in USSDM. Only most-recent data and objective analyses are used in the current MERCURY-2 design; selected past data will be maintained in phase-2 for comparison to numerical predictions.

The pressure height field from the synoptic objective analysis will be further analyzed using a slope analysis procedure similar to that employed to identify orographic features. Regions in the height fields having slopes above a specified value will be stored in an analysis feature list similar to the terrain feature list. This list will provide a rudimentary representation of synoptic features that may be expected to dominate or interact with mesoscale effects in the region of interest. This procedure for automatically analyzing objective heights, if successful, will also be employed for predicted heights.

Climatological data will be represented in the second phase of MERCURY-2 as point data for given locations. Details of the form of the climatological data to be used have yet to be determined.

3 4 DEVELOPER INTERFACE

The design of the MERCURY-2 developer interface combines the general layout and menu structure of the MERCURY-1 interface with the map display facilities provided by the USSDM package. The latter are based on the Graphics Kernel System (GKS) portable graphics drivers, which also underlie the NCAR Graphics package. MERCURY-2 is being designed to support 8 bit color graphics, in order to allow the use of color for displaying land use regions and derived meteorological products (e.g. contoured height fields).

3.5 DATA ANALYSIS SYSTEM

The organization of the MERCURY-2 data analysis system is shown in Fig. 7. The system employs mesoscale objective analysis, approximate diagnostic models, heuristic rules, and qualitative modelling to estimate the values of required variables under different conditions of terrain complexity and data availability. The modules implementing these different analysis strategies each have direct access to the data representations, and operate independently from each other. The analysis module or modules to employ in a particular situation are determined by a metarule base, which receives the request for data from the client TDA interface. The metarules evaluate each data request with respect to the type of data required, the availablity of data, and the terrain in the vicinity of the point for which data are required. By selecting the analysis module or modules to answer each request, the metarule base effectively directs the flow of control through the data analysis system. The use of metarules for control considerably simplifies the structure of the system, and increases the autonomy of each of the analysis modules (cf. Clancey and Bock, 1988).

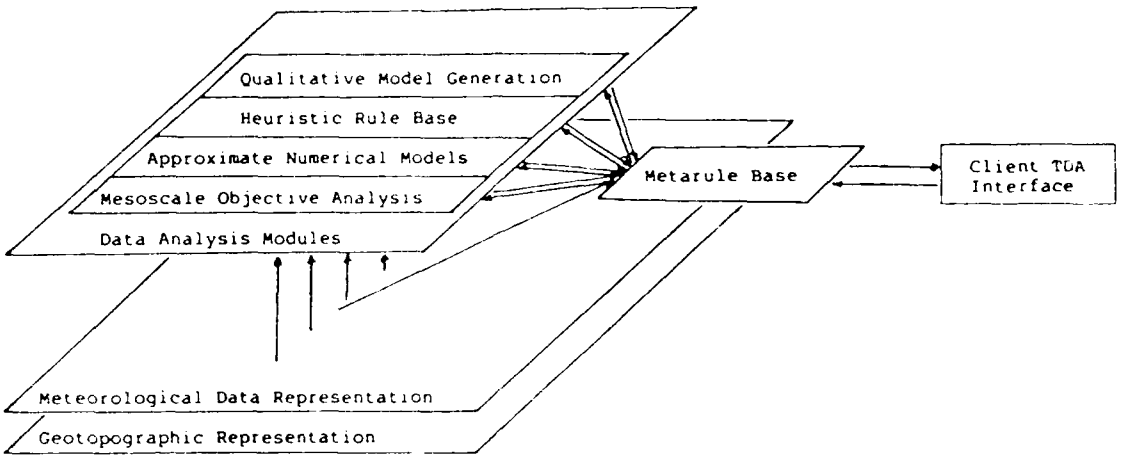


Figure 6. Organization of the data analysis system. Metarules are used to regulate the flow of control to the various data analysis modules. All modules have independent access to the data and geotopographic representations.

The action of the metarule base in evaluating requests and directing the flow of control can be illustrated by an example. If wind velocity data are required for a location near a coastline, for example, objective analysis may be employed if the required values can be obtained by interpolation from coastal stations. If the data available from coastal stations are too sparse for reliable interpolation, however, a better estimate of the coastal surface wind field may be obtained from an approximate numerical model for sea breeze penetration. In situations in which a sea breeze is unlikely, or in which the sea breeze may be influenced by, for example, an off-shore low-pressure system, qualitative modelling may be required to represent the interaction between the phenomena contributing to the wind field. The conditions under which these different strategies are likely to be useful are encoded as metarules, which then function to select the modules implementing the appropriate strategy for activation in each particular situation.

Specification of the functionality of the diagnostics and heuristics modules, and of the metarule base, has started, with the Los Angeles basin sea breeze as the initial phenomenon of interest. The sea breeze velocity V(r) as a function of radial distance r from the coast has been parameterized by the Fermi function:

$$V(r) = V_0/[\exp[w(r - r_0)] + 1],$$

where r = 0 is taken to be the coastline, V_0 is the velocity at the coastline, and V(r) is assumed to be radial. The penetration depth r_0 is assumed to vary diurnally as a Gaussian:

$$r_0(t) = r_0(max) \exp -[(t - t_0)/\tau]^2,$$

where $r_0(max)$ is the maximum penetration depth and t_0 is the time of maximum penetration. The parameters V_0, $r_0(max)$, and t_0 vary seasonally, and are assumed to be included in the climatology database. The 1/e length parameters w and τ are assumed to be constants, and are obtained by fitting data.

Diagnostic models for correcting wind speed for terrain roughness, and for correcting temperature and pressure for elevation, will be obtained from ASL. Diagnostic models for sensible heating, and for mountain breezes, will be obtained from ASL or developed as parameterizations.

Data source evaluation functions similar to those developed for the MERCURY-1 DEM will be used to evaluate data sources for representativeness of temperature, pressure, humidity, and precipitation variables. It is anticipated that different exponential decay parameters will be needed for different variables. Values of these parameters will be obtained by fitting appropriate data.

REFERENCES

Campbell, D. and R. Rew, 1988: Design issues in the UNIDATA local data management system. *Preprints of the Fourth International Conference on Interactive Information and Processing Systems for Meteorology, Oceanography, and Hydrology*. AMS. pp. 208-212

Cavendish, C., S. Kirby, and G. McWilliams, 1988: A knowledge representation scheme for surface feature and terrain elevation data: With special application to meteorology This Volume.

Clancey, W and C. Bock, 1988: Representing control knowledge as abstract tasks and metarules. In: L. Bolc and M. Coombs (Eds) *Expert System Applications*. Berlin: Springer. pp. 1-77.

Coombs, M. and R. Hartley, 1987: The MGR algorithm and its application to the generation of explanations for novel events. *Int. J. Man-Machine Studies* 27: 679-708.

Coombs, M. and R. Hartley, 1988: Explaining novel events in process control through model generative reasoning. *Int. J. Expert Systems* 1: 87-109.

Coombs, M., C. Fields, and G. McWilliams, 1988: *Artificial Intelligence Methods for Optimizing the Use of Meteorological Databases: Recommendations for Implementing the MERCURY System*. Final Report, Contract #DAAD07-86-C-0034, WAO #87-2.10-5, U.S. Army .mospheric Sciences Laboratory Technical Report #ASL-CR-88-0034-2.

Dyer, R., 1987: Expert systems in weather forecasting and other meteorological applications. *AI Applications in Natural Resource Management* 1: 19-24.

Fields, C., 1988: *Artificial Intelligence Methods for Optimizing the Use of Meteorological Databases: Architecture of the MERCURY System.* Final Report, Contract #DAAD07-86-C-0034, WAO #88-2.10-2, U.S. Army Atmospheric Sciences Laboratory.

Fields, C., M. Coombs, and R. Hartley, 1988: MGR: An architecture for problem solving in unstructured task environments. In: Z. Ras and L. Saitta (Eds) *Methodologies for Intelligent Systems, III.* Amsterdam: Elsevier pp. 40-49.

Hansen, F. V., 1984: Tactical smoke and chemical models. Technical Report, U.S. Army Atmospheric Sciences Laboratory.

Soderlund, C., 1988: A TCP remote message passing service. *Memoranda in Computer and Cognitive Science* MCCS-88-130, Computing Research Laboratory, New Mexico State University.

Young, K. C., 1987: An adaptive learning system applied to the problem of forecasting convective precipitation. Preprint, Institute of Atmospheric Physics, University of Arizona.

Applying Artificial Intelligence Techniques to the GIS Data Acquisition Problem

Robert F. Richbourg
Office of Artificial Intelligence Analysis and Evaluation
United States Military Academy
West Point, New York

Abstract

Geographic information systems (GIS) are capable of automating many of the analytical functions performed (manually) by terrain analysts. As with any computer system, the GIS software can only be utilized when an appropriate digital representation of the terrain to be analyzed is available. Currently, most digital data sets are hand-crafted, a tedious and error-prone process. In this paper, we discuss an artificial intelligence based approach designed to assist the human in the creation of digital terrain data. The paper describes the Digitized Overlay Object Recognition (DOOR) system which applies an automatic programming technique to a character separation and recognition problem. By relying on the system, a user can program stroke-based character recognition procedures that can be applied to digital images of engineering factor overlays, resulting in the creation of digital topographic data sets appropriate for a specific GIS, the TerraBase system. Use of the system increases data set accuracy and decreases the time required for data set production.

1. Introduction

The TerraBase system [Ref. 1,2] is a military terrain information system that has been developed by the Computer Graphics Laboratory (CGL) within the Department of Geography and Computer Science at the United States Military Academy, West Point, New York. TerraBase is designed to support U. S. Army engineer topographic units in the performance of terrain analysis tasks including the production of tint overlays, line of sight profiles, perspective terrain views and cross-country mobility overlays. The system is very capable and has generated a favorable reaction from a growing user community. However, as is the case with all geographic information systems (GIS), the functionality of the TerraBase system is limited by the availability of high quality, digital terrain data. Currently, only elevation data is widely available in digital form.

The TerraBase system includes a module called Digidata that allows the manual creation of digital data from analog sources. The DMA publishes an analog Tactical Terrain Analysis Data Base (TTADB) in the form of engineering factor overlays including vegetation height, vegetation type, surface configuration, and the soil types overlays. An example of a surface configuration overlay is shown in Figure 1. The actual TTADB overlays are approximately 17.5" X 22" and are drawn on clear acetate so that they can be placed on top of a standard map sheet. The Digidata program interfaces a digitizing tablet to an EGA resolution computer screen so that the TTADB data can be captured by tracing (with a digitizing puck) the TTADB areal boundaries on the digitizing tablet. Once these boundaries are entered, the character code associated with each area can be entered from a computer

keyboard. (Dependent upon the overlay type a one to three character code is associated with each region.) The manual crafting of digital data in this manner has proven to be tedious, labor-intensive and error-prone. Estimates for the time required to digitize the overlays associated with a single map sheet vary from 50 to 150 man-hours, depending on overlay complexity and the skill of the operator (the overlay shown in Figure 1 is of moderate complexity).

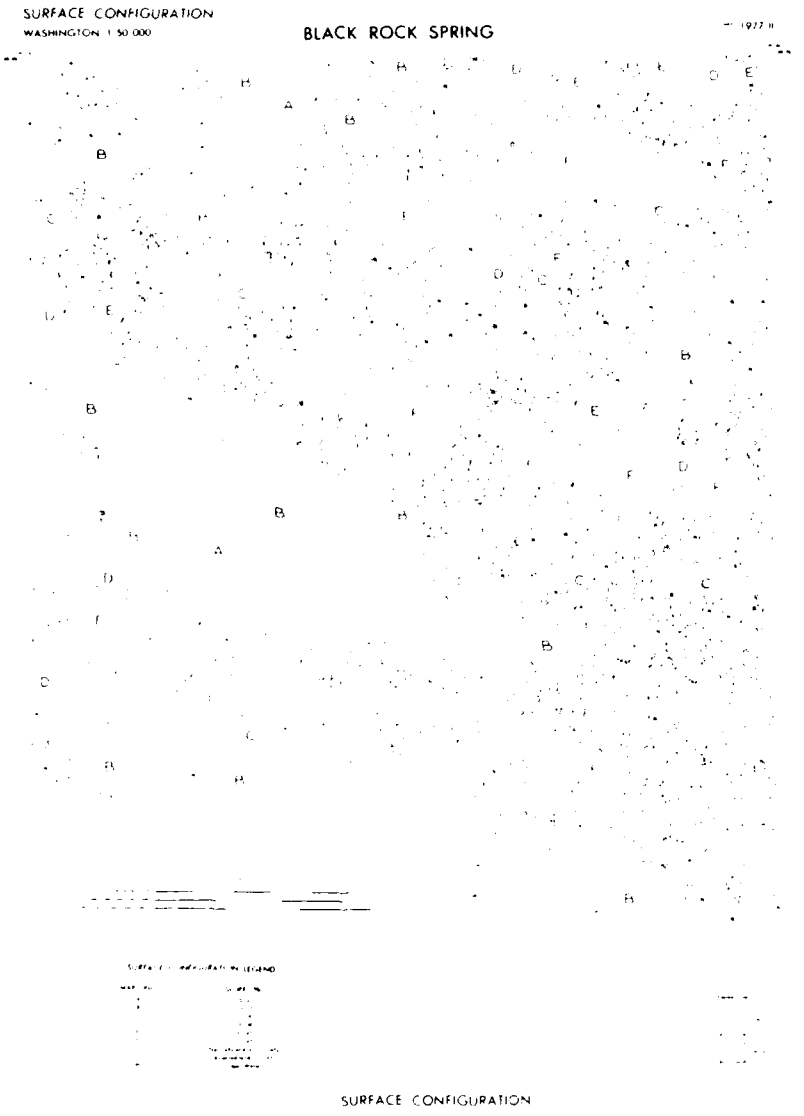


Figure 1. A Surface Configuration TTADB Engineering Factor Overlay

An experimental system is being developed in order to shift the labor intensity of the digital data capture process from man to machine. The first requirement of this system is that the digitizing puck and tablet be replaced with a digital imaging camera system. The camera can be used to capture a digital image of the entire overlay, to which character recognition techniques can be applied. If a system can recognize all the characters in the image, store their locations internally, and erase the characters from the image, then only the area boundaries

will remain. This process would complete the two steps effected by the Digidata program; the area boundaries would be "traced" and a character code would be associated with each area.

2. Design Constraints

Many character recognition techniques have been developed [Ref. 3,4,5,6]. However, some of these techniques do not apply well to the TTADB character recognition problem for several reasons, many of which are evident in the overlay of Figure 1. Multiple sizes of characters occur on single overlays. Character fonts differ between overlays. Individual character groups may be rotated (written at an angle or vertically) on some overlays. While most character groups are inside the area they describe, some are not and "leader lines" are used to indicate the associated area(s). The character locations "float"; unlike text, the horizontal and vertical spacing between characters and between character groups is non-uniform. The area boundaries contain many character-shaped portions which must not be confused with the characters themselves. Thus, as well as recognition, the technique must also be able to separate characters from a potentially confusing, irregularly shaped graphical background.

There are also system considerations which impact on the choice of a specific character recognition technique. The TerraBase system is designed to be hosted on machines that are commonly available to Army topographic units. For compatibility with TerraBase, the character recognition software should also execute on an MS-DOS PC/AT compatible machine. Thus, no special purpose hardware can be depended upon and both time and space resources must be conservatively managed. Further, software maintenance issues encourage that the recognition software be delivered in the same programming language as the extant TerraBase system.

Due to these considerations, a stroke-based recognition procedure was chosen for the implementation. These methods are more easily adapted to scaling and rotation changes than are the matrix-like, two-dimensional pixel pattern matching techniques. They also allow a hierarchical classification of the characters to be considered for recognition by using the strokes common to different characters as grouping keys. The stroke-based methods also seem more applicable and efficient in view of the floating text problem.

Because execution time is a primary concern, the system should provide some way for the user to specify those characters that should be regarded as candidates for recognition. Thus, the user should be able to load or unload character definitions from libraries as required. This effects a human-assisted pruning mechanism where only those characters that appear in the image at hand are considered by the recognition software, greatly reducing processing time and the potential for error. Further, the character definitions should be very specific, not only to speed execution but to reduce errors of commission as well. (As a design choice, errors of omission are preferable to errors of commission.) A final consideration is that the system should be able to efficiently recognize characters of different fonts, some of which may not be strictly defined (as in the case of hand-written characters). For this reason, our approach has been to make the system user-programmable. Thus, a user can not only reference symbol definitions from an extant library, but can also create new symbol definitions as the need arises.

3. The DOOR System

Above are many of the considerations that have impacted on the development of the Digitized Overlay Object Recognition (DOOR) system being developed at West Point. The system relies on a stroke-based, automatic programming [Ref. 7] (in the sense that FORTRAN was automatic programming during the 1950's) approach to character recognition. The DOOR system takes as input a binary digital image which has been processed by convolution [Ref. 8] and adaptive thresholding algorithms. Given this input, the user has two available options. Symbol definitions can be loaded (individually or in sets) from libraries or new definitions can be created by using the automatic programming capabilities of the system. If extant definitions are to be used, the user simply reads them in from disk to create a definition set and then invokes the (stand-alone) recognition software. A more interesting case arises when new definitions are to be created.

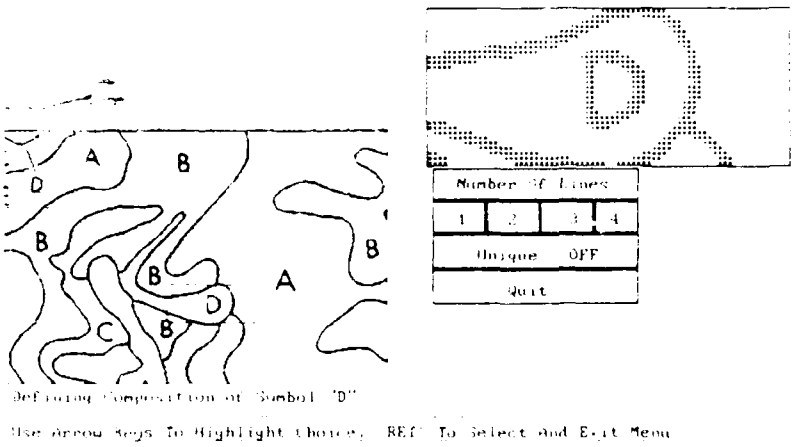


Figure 2. Programming A Symbol Definition

The left side of Figure 2 shows a small portion of a digitized factor overlay image as provided to the DOOR system. There is a thin-lined rectangle drawn around the letter "D" in the image. This is a user designated area which is enlarged and displayed in the top right portion of the figure. Here, individual pixels are distinguishable. The "blow-up" area is designed to assist the user in programming the system. This programming is accomplished by allowing the user to first specify important points (pixels) in the image of the character. Then, the connections between those points representing the strokes required to describe a symbol (or character) must be specified.

As an example, there are two important points used in writing the letter "D". One point is at the top left of the letter (the vertex of an inverted L-shaped line intersection)) and one point is at the lower left of the letter (the vertex of an L). These two points are connected once by a line segment and once by a half circle, opening to the left. This is the type of description provided to the system as programming.

In Figure 2, the menu entitled "Number Of Lines" is asking the user how many lines intersect at a point about to be specified. The user should select "2" since both points important in making a "D" feature the intersection of two line segments.

(This information will be used to prune candidates during the recognition process.) The user must then move a cursor to the pixel in the blow-up area that represents the intersection point. Once both of the important points for the letter "D" have been specified in this manner, the user is asked how the points should be connected. Menu options are used to select one linear connection and one circular connection between the two points (as described above). Provided with this description of the character "D", the DOOR system then creates the procedure calls and parameters necessary to apply the user-supplied definition. The parameters are adjusted so that a variety of "D" images can be recognized from the single description and this adjustment is based on the characteristics of the single "D" image used as the basis for programming. The adjustments allow for some rotation, scaling and, of course, translation of "D" images.

Similar definitions can be created for each symbol to be recognized in the input image. New definitions can also be intermixed with definitions loaded from libraries. In Figure 3, the system is advising the user that definitions for the symbols "A", "B", "C", and "D" are loaded. The "Alone" label indicates that each definition is for a single character, not a character that is part of a two or three character group. The number of

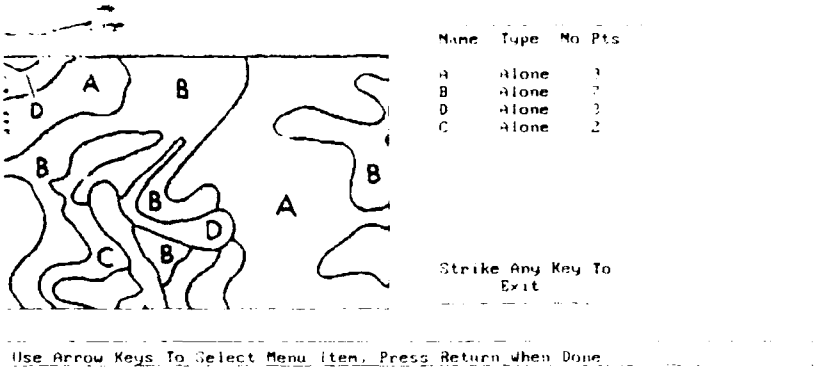


Figure 3. Symbol Definitions Known To The System

important points contained in each symbol definition is also listed. Note that the symbol definitions are specified by users during programming. The symbol definitions as described in Figure 3 were programmed by soldiers from a topographic unit. Why nine points were used to define "A" or three points were used to define "D" is unclear; however, the definitions work. The difference between specifying 2 or 3 important points in the letter "D" is not unlike that between the choice of using many individual (sequential) programming statements or a single programming statement inside a loop. Both methods can be used for the same purpose, but the latter is more concise.

Figure 4 depicts the result of applying the definitions to the portion of the overlay image as shown in Figures 2 and 3. Note that, except for the "B" on the right edge of the image, all of the characters have been rewritten in a standard font and a diamond-shaped marker appears by each letter. (The "B" not rewritten was not considered; it is in an "overlap" region of the image that would be processed on the next page.) The new font and diamond marker indicate those characters that have

been recognized by the system. The diamond marker designates the actual (logical) location associated with each character. Note that the intended location for the "D" in the upper left portion of the image, indicated by a leader line, has been correctly identified.

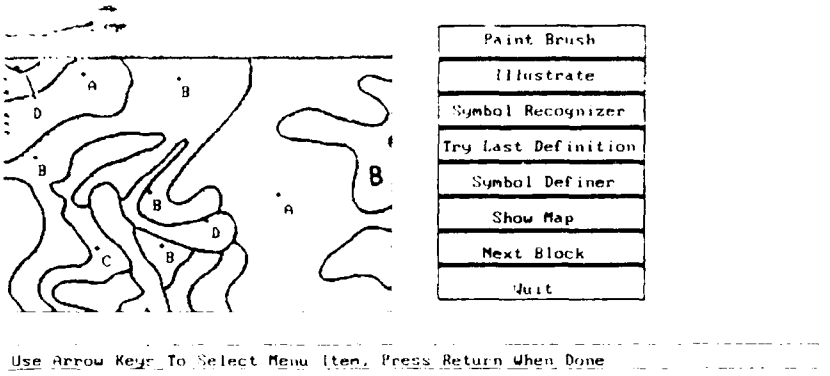


Figure 4. System Recognized Characters

4. System Usage

The DOOR system software includes four main components. The first component accepts as input a digital image produced by a 4096 X 4096, 8 bit deep (in intensity) CCD digital imaging camera system. A standard convolution technique is applied to this 16MB image to effect noise suppression and edge enhancement. Then, an adaptive thresholding algorithm is used to reduce the 8 bit deep input file to a binary representation (1 bit deep), yielding a 2MB file containing the sharpened image.

This enhanced, binary image is used as input to the second main component of the system, the symbol definition program as described in the previous section. The end result of this system component is the creation of a set of symbol definitions useful in recognizing symbols on a specific overlay image. The symbol definition set and the same enhanced, binary image are used as input to the third system component, the recognition component which produces two outputs. One output is a data file containing the symbol identifiers and their locations in the image. The second output is a processed image file where all of the recognized symbols have been removed, leaving only area boundaries and any unrecognized characters. If 100% of the characters were recognized, only area boundaries appear in this image file.

The system has never achieved 100% recognition, necessitating the final system component, a digital image editor. The editor is used to correct any errors produced by the recognition software. Symbols can be added to, deleted from, or modified within the set of recognized characters. The editor can also be used to repair noise corrupted portions of the image itself. A final editor capability causes the creation of the data files as required by the TerraBase Digidata program. The Digidata program expects a 900 X 900 pixel data file where each pixel is color-coded to represent its associated character description. This is a very time consuming operation since the entire overlay image must be registered, color-coded, and

compressed from (approximately) 4096 X 4096 pixels to 900 X 900 pixels. To register the image, the corners of the overlay must first be located within the image (generally, there is some amount of "white space" around the outside of the overlay that also appears in the digitized image). Often, the overlay image requires some slight rotation so that it becomes orthogonal with the screen coordinate system. Also, the overlays are either rectangular (for 1:50,000 scale overlays) or pieces of two-dimensional conic sections (for 1:250,000 scale overlays). This shaping requires a non-uniform compression into the 900 X 900 square space.

5. Conclusions

A prototype, experimental version of the DOOR system has been used by Army soldiers from several different units to create digital data for use with the TerraBase system. Members of the 649th Engineer Battalion (TOPO) of the 18th Engineer Brigade used the system to support the REFORGER 88 exercise in Germany. The system has also been used by soldiers from the 30th Engineer Battalion, Fort Belvoir, Virginia and from the 557th Terrain Detachment, 8th Army, Korea. On the average, the system produces at least a 10 fold reduction in the man-hours required to create digital data. Recorded character recognition rates ran from a low of 62% to a high of 96% correctly identified characters. The system was not as efficient in recognizing occurrences of leader lines meant to indicate character locations. These were correctly processed in only about 50% of the cases. We note that the percentage of correctly identified characters is directly related to the skill with which the soldiers programmed the system. As the user-soldiers gain more expertise and more accurate libraries of symbol definitions are created, the percentage of correctly identified characters should increase.

The recognition software requires a large amount of machine processing time on the Zenith 248 (8 MHz clock) personal computer. Up to 10 hours may be required to process a single overlay image using this machine. (Note that this time is not included in the man-hours required to use the system since the recognition software executes with no human intervention.) Due to this large processing requirement, some users have bypassed the recognition step and used the image editor to enter all character descriptions. The editor is menu driven, allows mouse input, and is easy to learn and use. An overlay of average complexity can be entirely processed with the editor is less than 2 man-hours.

The DOOR system has demonstrated its utility. It eliminates many man-hours of manual labor and facilitates the production of accurate digital data. The time requirements of the recognition software can be greatly reduced by using an MS-DOS 386-based machine. Also, these requirements become less important when several 286-based machines can be used as dedicated processors. Regardless of the mode of usage, the DOOR system has proven its value in greatly reducing the tedious, error-prone nature of a time consuming task.

6. References

1. Ressler, E. K., "Ninety Per Cent Today; Activities of the USMA D/G&CS CGL", *Proceedings, Department of Defense Mapping, Cartography, and Geodesy Conference*, October 1987.

2. Ressler, E. K., "TerraBase -- Geographic Information System for Military Data", presented at the 1987 Army Topographic Conference, Washington, D. C., October 1987.

3. Cox, C. H. III, Coueignoux, P. and Blesser, B., "Skeletons: A Link Between Theoretical and Physical Letter Descriptions", *Pattern Recognition*, Vol. 15, No. 1, pp 11 - 22, 1982.

4. Kahan, S., Pavlidis, T. and Baird, H. S., "On the Recognition of Characters of Any Font and Size", *IEEE Trans. PAMI*, Vol. 9, No. 2, March 1987.

5. Pavlidis, T., *Algorithms for Graphics and Image Processing*, Computer Science Press, Rockville, 1982.

6. Vlontzos, J. A. and Kung, S. Y., "A Hierarchical System for Character Recognition with Stochastic Knowledge Representation", *Proceedings, IEEE International Conference on Neural Networks*, San Diego, July 1987.

7. Barr, A. and Feigenbaum, E. A., *The Handbook of Artificial Intelligence*, William Kaufmann, Los Altos, 1982.

8. Winston, P. H., *Artificial Intelligence*, Addison-Wesley, Reading, 1984.

AN EXPERT SYSTEM FOR MINEFIELD SITE PREDICTION

Jonathan W. Doughty, Anne L. Downs,
Michael J. Gillotte Jr. and Stephen A. Hirsch
PAR Government Systems Corporation
Reston, Virginia 22090, U.S.A

ABSTRACT

The design and implementation of a Minefield Site Prediction Expert System (MSPES) is described. The ultimate goal of the system is to emulate the role of a terrain analyst and combat engineer in predicting likely minefield sites. The major components of the system are the inference system, the geographic information system, and the user interface. The inference system is driven by a goal-directed backward chaining mechanism. The geographic information system (GIS) is based on quadtrees. The user interface is window-based, and uses an object-oriented graphics package.

This paper discusses the system architecture, the issues that arose during the design and implementation of the prototype system, and the resolution of those issues. In particular, the tradeoffs between making decisions in the GIS component and in the inference system component are discussed.

1. INTRODUCTION

PAR Government Systems Corporation (PGSC) is currently under contract to the U.S. Army Engineer Topographic Laboratories to develop the Minefield Site Prediction Expert System (MSPES). The purpose of this system is to automate some of the functions performed by the terrain analyst and the combat engineer in the determination of potential minefield sites. The factors used in minefield site prediction include terrain information, such as cross country mobility information; mine/countermine warfare doctrine, as found in military training manuals; and battlefield situation or enemy intention knowledge, as supplied by battlefield intelligence. The first phase of the MSPES development was completed in January 1988 and resulted in a prototype system on a Sun 3/160 color workstation under the Unix operating system. Phase II, currently being performed, is an expansion of the prototype system, including the use of the X Window System and enhancement to the rule base. Phase III, scheduled to begin in December 1988, will center around the porting of the system to a DEC VAXstation II/GPX under the VMS operating system.

Developing an expert system for predicting potential minefield sites involves elements of military terrain analysis, which in turn encompasses both geographic analysis and military doctrine. The system must therefore embody a geographic information system, for handling the terrain information; an inferencing mechanism, for

coordinating rules about how doctrine exploits the terrain information in making minefield site predictions; and a user interface, with which the analyst working in this domain feels comfortable.

This domain, the prediction of likely minefield sites, is particularly suited to expert system development because it is a well defined and bounded problem. Most of the terrain and geographic analysis which is involved in the prediction of minefield sites can be codified into a set of rules. For example, troops will not find antitank mines in a forested area with stem diameters greater than twenty five centimeters and a stem spacing of less than two and one half meters. Nor would antitank mines be found on a slope greater than a forty five degree angle. Mine/countermine warfare doctrine and much of the battlefield environment also can be represented in an expert system. The doctrine applying to mine warfare in a certain geographic location, for example Europe, is fairly consistent regardless of the nationality of the forces. In other words, Soviet minefield doctrine and U.S. minefield doctrine are similar for European terrain and situation. With these factors established by knowledge engineering and research, the MSPES becomes an effort of system integration and rule base development. The following sections of this paper discuss the prototype MSPES and the direction of future MSPES development.

2. SYSTEM COMPONENTS OF PROTOTYPE

The Minefield Site Prediction Expert System consists of three components: an inference system, a geographic information system capability, and the user interface.

2.1 INFERENCE SYSTEM

The inference system used by the MSPES is ERS, the Embedded Rule-based System. This inference system, developed by PGSC, is a framework for the development of expert system applications. ERS is written in the C language and uses a goal-directed, backward chaining mechanism. It includes a consultation capability and facilities that allow rules to access, evaluate, and modify external sources of information.

The ERS consists of a rule base parser and an inference 'engine'. ERS rule bases are text files written in the ERS rule-base language. On startup, ERS reads a specified rule-base file, parses it, and compiles the rules into internal data structures. The internal data structures control the inferencing mechanism, which in turn directs the gathering of evidence in support of hypotheses. Figure 1 represents the inference system and the relationship between it, textual rulebases, and sources of external information.

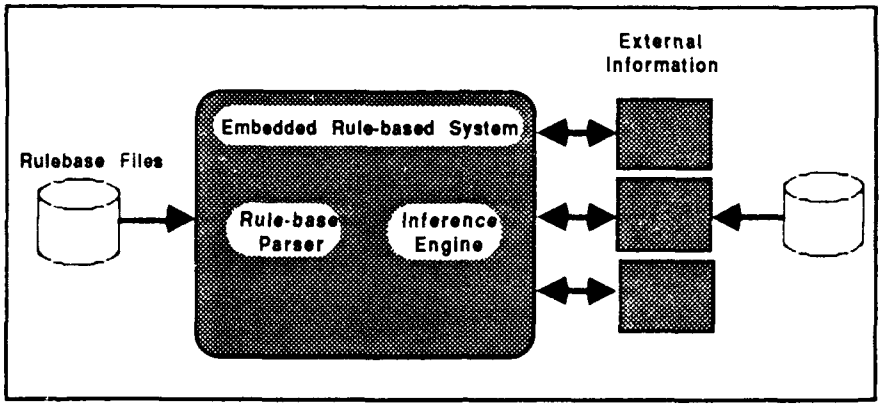


Figure 1 - Inference System, Rulebases, and External Information

2.2 GEOGRAPHIC INFORMATION SYSTEM

The geographic information system (GIS) capability used by the MSPES is the QUILT system, developed by the Center for Automation Research at the University of Maryland. QUILT is an experimental system that uses disk-file based, quadtree data structures to support the storage and retrieval of information about point, linear, and regional data. Figure 2 illustrates the transformation of a polygon into a quadtree.

The quadtree storage technique uses successive subdivision of a square area into quadrants to represent polygonal areas. Subdivision of quadrants takes place until each quadrant has a homogeneous value: it is either wholly inside or outside the polygon. Quadtrees are particularly useful in the performance of spatial operations such as point in polygon determination, polygon intersection, and union.

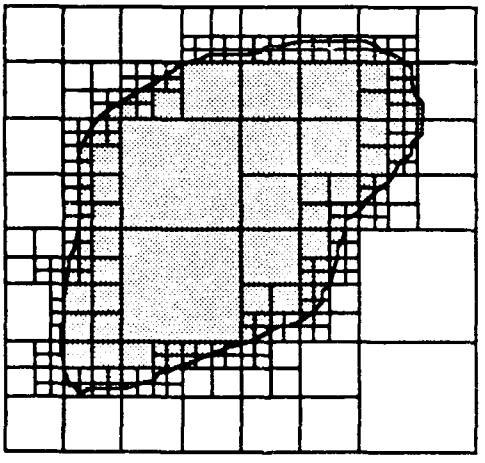


Figure 2 - The Transformation of a Polygon Into a Quadtree

QUILT consists of a kernel of core functions required for creating and traversing quadtrees, and a collection of programs

that use these kernel functions to perform various application specific manipulations of the quadtree data. Quadtree manipulations that are provided as part of the QUILT system include extracting a subset of the areas with specific values, changing all areas with a value to a new value, finding all areas that are within a specified distance of areas with a particular value, determining the intersection of areas in two maps, and determining an attribute associated with a specified map location.

Figure 3 represents the geographic information system capability used by the MSPES and the relationship between the QUILT kernel, the application software that uses the kernel, and the database of quadtree data structures QUILT maintains.

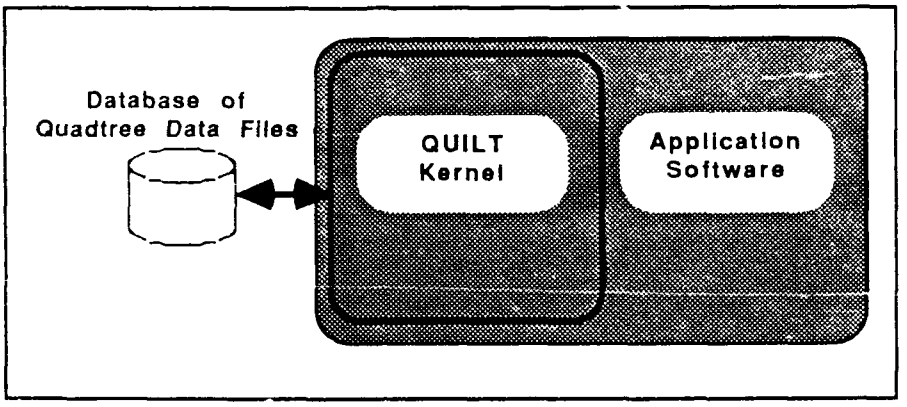


Figure 3 - GIS Capability, Application Software, and GIS Database

2.3 USER INTERFACE

The user interface of the prototype MSPES uses the SunView 1.0 graphics package provided by Sun Microsystems Inc. SunView 1.0 is an object-oriented user interface toolkit that is part of the Unix kernel of the Sun operating system. SunView provides window based application interfaces using command buttons, menus, graphic canvases, and text windows. SunView notifies applications when various window events occur, such as cursor movement, mouse button clicks, and keyboard responses.

Applications that use the SunView package to create their user interface typically use an object-oriented, event driven paradigm. An application defines its user interface by creating a frame and filling that frame with user interface components. These components may include objects such as buttons that users can select, canvases that the application can draw on, and text windows to display prompts and user responses. As the application program creates each user interface component, the program nominates a function as callback routines that will handle events in that component. The application then invokes the Notifier. The Notifier waits for the user to generate events within the application window frame, identifies the user interface component associated with the event, and invokes the function that had been nominated as the component's callback function. Figure 4 represents the interaction between SunView and an application.

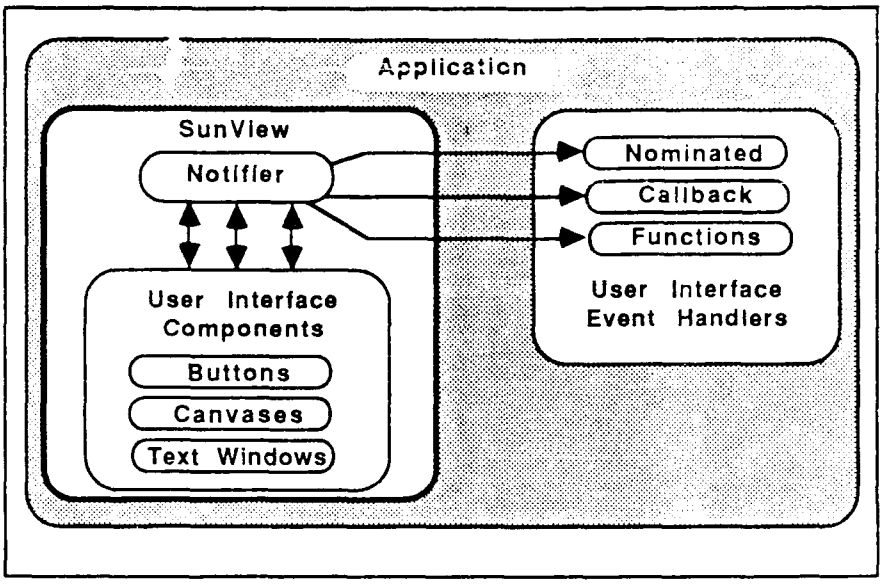


Figure 4 - SunView - Application Interaction

3. COMPONENTS OF THE KNOWLEDGE BASE

The MSPES Knowledge base consists of terrain data, spatial reasoning, rule bases, and geographic information system primitives.

3.1 TERRAIN DATA

The prototype MSPES uses two kinds of terrain data: Cross Country Mobility (CCM) information and Line of Communication (LOC) information. CCM data specifies whether an area permits a vehicle to traverse it easily, with difficulty, or not at all. This is represented by the CCM values "Go", "Restricted", "Slow", "Very Slow", "No Go", "Built-up", and "Open Water". The CCM data used by the prototype MSPES is derived from the Defense Mapping Agency's Cross Country Mobility model. This CCM model factors in parameters for soil type, vegetation cover, and slope or surface configuration as they pertain to the movement of a specific type of vehicle. The Phase II MSPES uses mobility information from the Condensed Army Mobility Model which uses additional terrain factors and more specific factor inter-relationships. LOC data provides a linear network of traversable paths. Roads, railroads, and navigable rivers are three Line of Communication feature categories.

3.2 SPATIAL REASONING

Spatial reasoning in the context of MSPES involves several measures of space to determine the likelihood of a minefield placement. Space in MSPES is partitioned via a quadtree data structure as described above. Each node in the quadtree represents a specific location which is measured in terms of proximity, area, and characteristics. The proximity measure provides a distance

from each location to the nearest segment of a road network. The area measure provides the size of each homogeneous quadtree node. Characteristics of each location indicate CCM category and 'canalization'. The CCM category provides a measure of mobility for a particular vehicle through an area, considering the soil type, vegetation cover, and slope of the area. The term 'canalized' refers to movement across the terrain being channeled into narrow paths, for example, into a canyon. Canalization is a shape measure of "Go" areas: where the area is narrow and thus constrains movement, the area is said to be canalized.

The spatial measures utilized in MSPES represent relatively lower level forms of spatial reasoning. Implemented as GIS primitives, these measures are invoked by rules within the inferencing component. A higher level form of spatial reasoning is performed in the inferencing component, where minefield doctine is represented. Minefield doctrine thresholds, criteria, constraints, etc., to a large degree, focus on spatial aspects of locations regarding their likelihood of being a minefield site. Spatial reasoning at this level might consist of the composite of reasoning performed at lower levels. For example, a location is a likely minefield site if it is in a canalized area and is within 'X' distance of a road and has a CCM category of "Go".

Spatial reasoning in the prototype system is fairly rudimentary. The reasoning capability will be expanded in Phase II and is discussed below.

3.3 RULE BASE

The rule base of the prototype system incorporates rudimentary knowledge about terrain factors and how these factors influence the location of a minefield site. The main effort of Phase I was to integrate the geographic information system capability, the inferencing mechanism, and the user interface rather than to develop an extensive rule base. Phase II focuses on rule base enhancement and in developing a "complete" expert system for minefield site prediction. The discussion below focuses on the prototype rule base.

The purpose of the rule base is to determine the likelihood of a minefield being present at a certain location. Four categories of likelihood are assigned: "Very Likely", "Likely", "Possible", and "Unevaluated". Each one of these categories is represented by a separate rule base goal node.

The inferencing mechanism of the Embedded Rule-based System requires a rule base of simple syntax and structure. A rule is essentially an IF-THEN statement that relates a single or set of antecedent conditions, the IF part, to a single concluding assertion, the THEN part or consequent, with some weight of evidence. Each antecedent condition and consequent has a degree of belief associated with it, specified initially by the rule base as a prior value. Figure 5 is an example of one of the rules in the MSPES prototype rule base.

```
node likely
  text desc
    " the current region is a likely minefield site"
    explanation
    " Likely minefield sites are where movement is canalized,
    and/or a road is present, and CCM is possible. "
  inference
    prior  -5.0
    bayesian antecedents
      (
      ccm_possible  pw 5  nw 0
      canal_area  pw 10  nw 0
      road_loc  pw 10  nw 0
    control
      context of unevaluated int min 0.0
      context of not_likely   int min 0.0
      context of ccm_possible int 0.0 max
      goal
```

Figure 5 - An ERS Rule Base Rule

The collection of rules forms an inference network which is a directed graph of nodes and links. Nodes in the inference net represent individual antecedent conditions, consequent assertions, or context conditions for the rules and may be categorized as goal nodes, hypothesis nodes, or evidence nodes. Figure 6 is a representation of the inference network of the prototype rulebase.

An evaluation of "Unevaluated" indicates that an area has a mobility category of "No Go" or that there is not enough evidence supplied by the input data to assign one of the other categories. The evaluations of "Possible", "Likely", and "Very Likely" result from increasing probability that a location is a mine site based on the terrain characteristics of the location and adjacent areas.

The following assumption is made regarding the knowledge that effects the organization of this inference net: if an area is a "Very Likely" minefield site, it is also a "Likely" minefield site which in turn indicates that it is also a "Possible" minefield site. In other words, except for unevaluated areas such as urban centers and "no-go" areas, all remaining terrain is considered a "Possible" minefield site with subsequent category refinement into "Likely" and "Very Likely" depending on the weights of additional evidence.

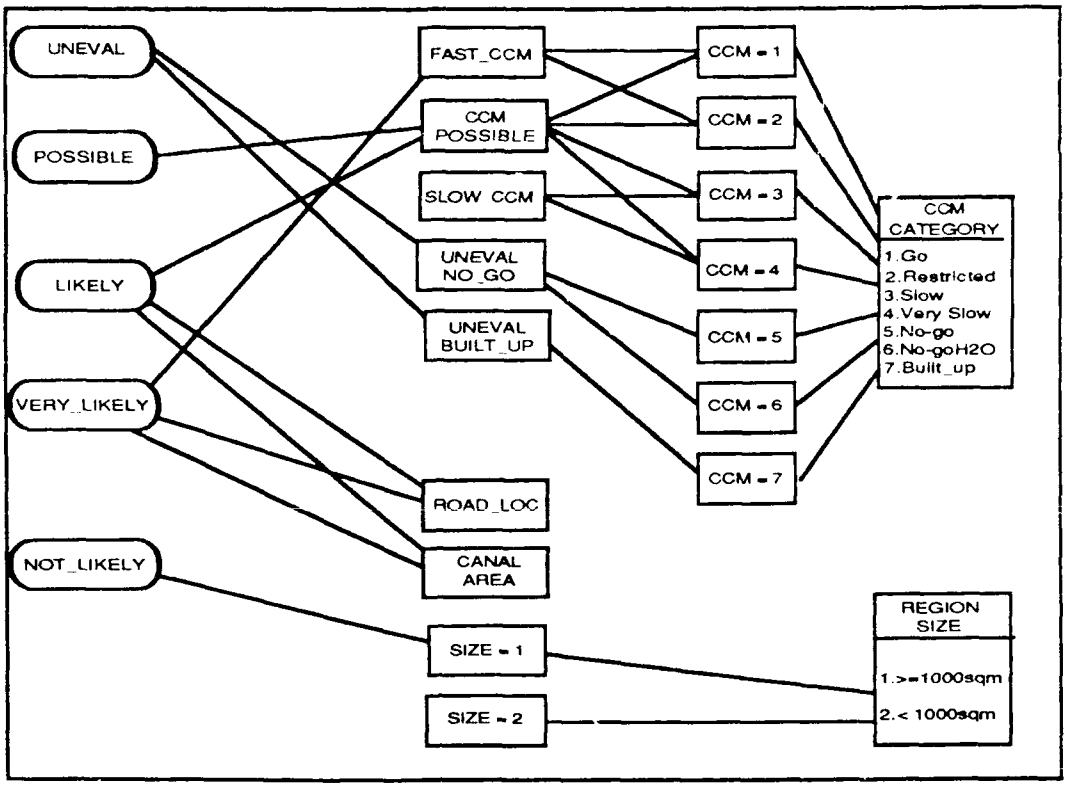


Figure 6 - The Prototype Rulebase Inference Network

3.4 GEOGRAPHIC INFORMATION PRIMITIVES

During the operation of the MSPES, eight different, single-purpose processes are used to access data maintained in the QUILT System's quadtree data structures. These processes are referred to as GIS primitives. Four of these primitives are used by the inference system to supply evidence of terrain characteristics at specified geographic locations. The remaining four primitives are used to drive application actions.

The four primitives used by the inference system provide information about characteristics associated with locations within terrain overlays. The GIS primitives that access this information answer the questions: "Is this location near a road?", "What is the CCM category associated with this location?", "How large is the homogeneous area associated with this location?", and "Is the area associated with this location 'canalized'?".

The four primitives that are used to drive application actions provide ancillary data used in system operation. One primitive provides a stream of locations of homogeneous CCM category areas to drive the inference system and updates the values of those locations with the evaluation that the inference system provides. Another provides information for the display of quadtree data. A third provides database update facilities and is used to permit analysts to over-ride the conclusions reached by the inference

system. The last primitive converts coordinates, entered through the user interface, to database coordinates identifying areas of homogeneous characteristics.

4. SYSTEM OPERATION

The MSPES is made up of the three loosely integrated components of the inference system, the geographic information system capability, and the user interface. In the prototype MSPES these components are realized as separate, cooperating processes in the Unix operating system environment.

The components were kept loosely integrated for several reasons. First, one or more of the components might have required replacement following the evaluation of the prototype MSPES. Loose integration of the components with well defined linkages to the others facilitates replacement of a component if that was deemed necessary.

Second, since the inference system and the geographic information system capability used "off-the-shelf" software components, keeping them as separate, communicating Unix processes avoided naming and addressing conflicts and simplified the integration process.

A third reason is related to the QUILT kernel which is designed to deal with a small number of simultaneously open quadtree disk files. By designing the system around geographic primitive processes that provide simple units of information, it was possible to avoid requiring the GIS capability to increase the number of simultaneously open files.

Finally, by designing the system as separate, independent processes, it is possible to take advantage of a degree of parallelism in the component's operation. In addition, separate processes makes possible the distribution of components among networked processors.

4.1 MSPES APPLICATIONS

Seven applications make up the prototype MSPES: Input Map, Create Manuscript, View Map, Explain Manuscript, Edit Map, Edit Rulebase, and Help. Each application has a separate user interface definition which nominates specialized and shared functions to control its operation.

Input Map allows the user to specify where in the file system an input data source may be found and what kind of overlay the data represent. This information is used to start a script of processing steps to convert the input data format into the GIS database format.

Create Manuscript permits the user to specify the Area of Interest (AOI) and the rulebase that should be used to evaluate that AOI. This information provides the inference system with what is needed for initialization. The application starts a GIS

primitive to derive coordinates that are then passed to the inference system to be evaluated.

View Map provides graphic display capabilities associated with the viewing of terrain overlays and minefield site prediction manuscripts.

Explain Manuscript permits a user to display a minefield site prediction manuscript and interactively make inquiries of the inference system. The possible inquiries include how the inference system evaluated manuscript locations, why certain rules are fired, request elaborations and explanations from the rulebase about it's rules, and see the evidence provided by the GIS. Figure 7 depicts the Explain Manuscript application's user interface, which is typical of that of all of the MSPES applications.

Edit Map allows a user to display terrain overlays and manuscripts and interactively modify their content.

Edit Rulebase provides text editing capabilities that permit rulebase files to be modified.

Help provides information relating to all the MSPES applications. Help is also available within each of the applications pertaining to their individual operation.

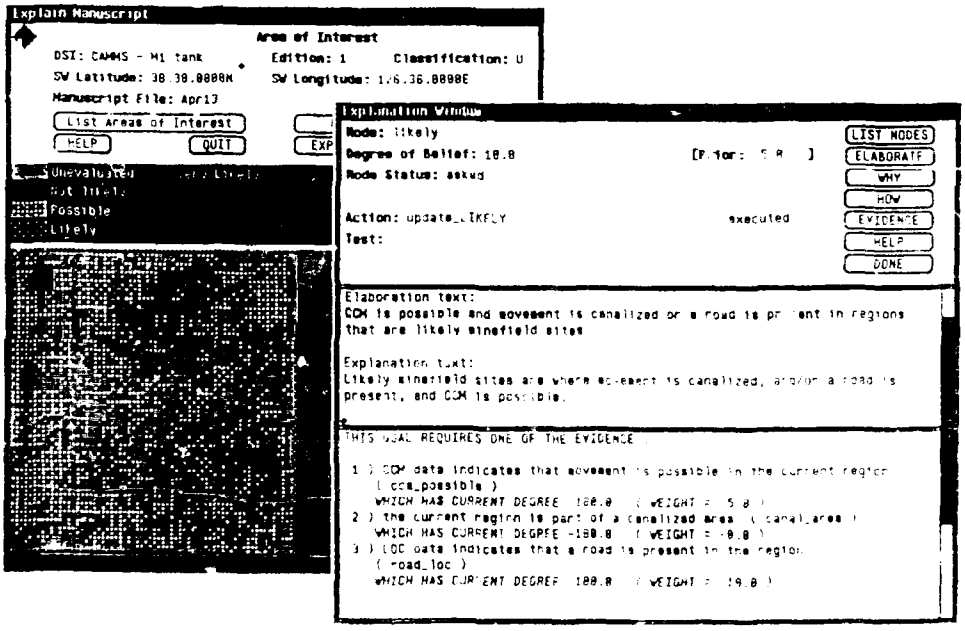


Figure 7 - The Explain Manuscript Application's User Interface

4.2 SYSTEM FLOW

The terrain overlays associated with an Area of Interest and a rulebase are the necessary inputs to start the process in which the MSPES inference engine may run. The textual rulebase that the system will process is read from a disk file and is compiled. The

compiled rules become the inference network for ERS. The inference network drives the process of gathering evidence for the various hypotheses about a location being a mine site.

Locations which are to be evaluated are specified to ERS by the Create Manuscript or Explain Manuscript applications. The Create Manuscript application gets its AOI locations from a geographic primitive, whereas the Explain Manuscript application gets its AOI locations from the analyst interactively. Because of the way AOI locations are identified, they are guaranteed to have an homogeneous CCM category.

The evidence in support of ERS's inferential hypotheses comes from GIS primitives. The primitive processes that ERS uses are started following the compilation of the inference network. The relationship of terrain characteristics relative to a location provide the evidence ERS uses as the basis for an evaluation of the likelihood of the location being a mine site. The evidence in support of the possible hypotheses is evaluated and the hypothesis with the highest 'score' becomes the evaluation for the specified location.

The Create Manuscript application sends this evaluation back to the geographic primitive that initially reported the location coordinates. This primitive updates the value associated with the location to reflect the mine site likelihood evaluation. Since the database file used for this purpose is never accessed by ERS, this evaluation does not bias later evaluations. The Explain Manuscript application reports the evaluation and related rulebase information to the analyst via a window-based interface to ERS. Figure 8 represents the various components referred to in the previous discussion and their interactions.

5. PROTOTYPE EVALUATION

.1 GLOBAL / LOCAL PROBLEM

Spatial reasoning in the prototype system is partitioned between primitive processes and rules interpreted by the inferencing system. Spatial reasoning in the form of well defined measures or measures are computationally intensive have been implemented as primitives for the sake of processing speed. Minefield doctrine, on the other hand, has been implemented as rules. The representation and exploitation of minefield doctrine is more dynamic, requires global as opposed to local information, and employs composite reasoning. In addition, a significant capability of the system is to allow minefield analysts to add or modify minefield doctrine. This is much more easily done by analysts at the rule base level than at the primitive level.

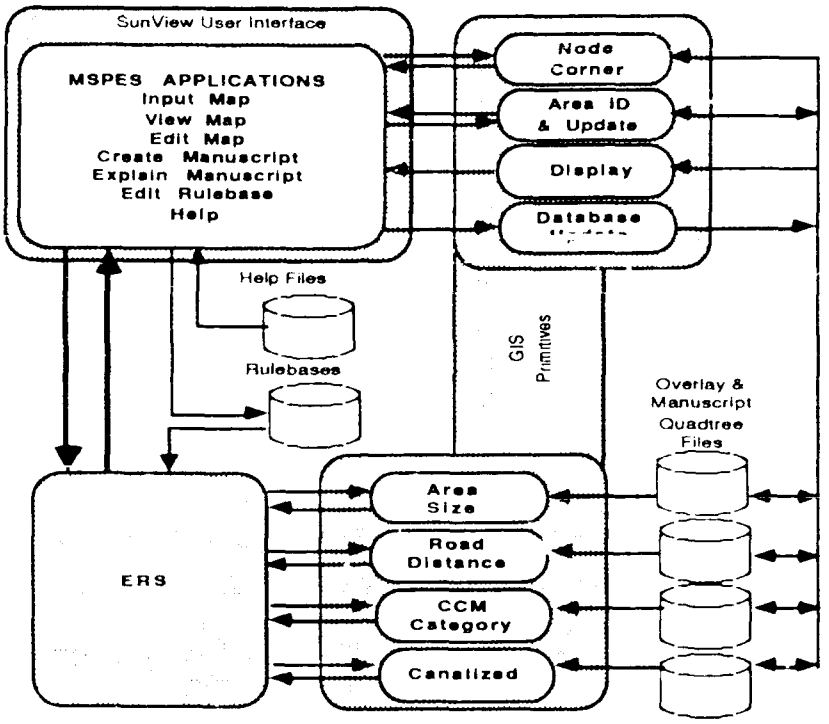


Figure 8 - Linkages Among MSPES Applications, ERS, and GIS Primitives

As the system is enhanced and new primitive requirements are identified, new primitive capabilities will have to be developed by personnel who are more intimately familiar with programming and the details of the system. One such activity is underway in Phase II; namely, the development of a more robust canalization primitive than was used in the prototype. The definition of a "canalized" area in the prototype MSPES is one in which movement is channeled into either a north-south or east-west direction. In the prototype, only local information could be exploited since the canalization primitive operated on quadtree representations of "Go" areas which subdivided homogeneous geographic areas into a series of quadtree nodes in accordance with the quadtree construction rules. The Phase II concept of canalization uses the "Go" areas skeleton and a minimum distance measure in an attempt to better reflect geographic reality. The enhanced primitive involves a preprocessing skeleton-ization function that exploits global information in identifying canalized areas.

Another activity under Phase II involves the incorporation of counter-mobility doctrine within the rule base. This will place additional requirements on spatial reasoning primitives and rule constructs including the ability to define an area of approach and composite this with other evidence.

5.2 RULES VS. PRIMITIVES.

As is typical with prototype development activities, the design must allocate processes to the appropriate component of the system. The foremost purpose of this allocation is to address functionality and secondarily to maximize efficiency. In the MSPES design this involved identifying what information needed to be incorporated into the rule base and which evidence could be supplied by primitive functions. Generally it is more efficient to run a primitive function which calculates some parameter than to have to derive this parameter through the interpreted inferencing process. Knowledge/ information that was required by the MSPES rule base was categorized into inferential and computational forms, and associated rules and/or primitives were allocated accordingly.

In addition to the allocation of primitives and rules, functions were identified as potential preprocessing steps, before the rule base and inferencing mechanism would actually be invoked. This can greatly increase the efficiency of the system. For example, in the MSPES environment an areal mask can be generated to screen out all "no-go" areas before starting the inferencing process. If a tank cannot move through an area, one need not be concerned with finding antitank mines there. This type of function reduces the number of areas that need to be investigated during the inference process. Because of the simplicity of the initial MSPES rule base in the prototype, there were no complex design issues relating to resource allocation. During Phase II, when doctrine and battlefield assessment become part of the enhanced rule base, the allocation issues relating to both rules versus primitives and doctrine versus computational efficiency will play a greater role.

6. SUMMARY

The MSPES is an example of linking expert systems technology and GIS capabilities to create an application requiring terrain analysis and inferences based on dynamic information. The minefield site prediction scenario incorporates elements demanding geographic and spatial analysis in addition to expert domain knowledge of mine warfare doctrine. This system provides the capability to work with these different elements and removes much of the processing burden from the analyst.

After successfully linking the inference mechanism, the GIS capability, and the user interface, the challenge was to allocate functions between the inferencing component and the geographic analysis component. These design issues and decisions would differ for applications other than site prediction depending on efficiency requirements and processing priorities. The MSPES achieves the allocation balance which best suits the goals of this system; that is to automate some of the functions performed by the terrain analyst and the combat engineer in the determination of potential minefield sites.

REFERENCES

Duda, R.O., Hart P.E., and Gaschnig, J., 1979: "Model Design in the PROSPECTOR Consultant System for Mineral Exploration" in Mitchie, D., Expert Systems in the Micro-electronic Age, Edinburgh University Press, Edinburgh.

Dillencourt, M., Doughty, J., Downs, A., 1988: Expert System for Minefield Site Prediction, (Phase I), ETL-0492.

Paterson, A., 1981: AL/X User Manual, Intelligent Terminals Ltd., 15 Canal Street, Oxford, UK 0X2 68H

Samet, H., 1984: "The quadtree and related hierarchical data structures", ACM Computing Surveys, 16, 2 , 187-260.

Samet, H., Rosenfeld A., Shaffer, C.A., Nelson, R.C., and Huang, Y-G., 1984: Application of hierarchical data structures to geographical information systems phase III, Computer Science TR-1257, University of Maryland, College Park, MD.

Samet, H., Rosenfeld A., Shaffer, C.A., Nelson, R.C., Huang, Y-G., and Fujimura, K., 1985: Application of hierarchical data structures to geographical information systems phase IV, Computer Science, University of Maryland, College Park, MD.

Shaffer, C.A., Samet, H., Webber, R.E., Nelson, R.C., and Huang, Y-G., An implementation for a geographic information system based on quadtrees, Tutorial Lecture notes #25, SIGGRAPH 86.

Rosenfeld A., Samet, H., Shaffer, C.A., and Webber, R.E., 1982, Application of hierarchical data structures to geographical information systems, Computer Science, TR-1197, University of Maryland, College Park, MD.

Rosenfeld A., Samet, H., Shaffer, C.A., and Webber, R.E., 1983, Application of hierarchical data structures to geographical information systems phase II, Computer Science TR-1327, University of Maryland, College Park, MD.

van Melle, W., 1980, A Domain-Independent System that Aids in Constructing Knowledge-Based Consultation Programs, Ph.D Dissertation, Report No. STAN-CS-80-820, Computer Science Department, Stanford University.

NEURAL NETWORKS FOR THE REALISTIC BATTLEFIELD

by

Edward M. Measure
Jeff M. Balding

U.S. Army Atmospheric Science Laboratory, WSMR, NM 88002

ABSTRACT

Neural networks appear to have potentially important advantages over conventional computers for performing some computational tasks on the realistic battlefield. One method for testing the potential of such systems is the development of simulations of such tasks. In conjunction with development of a radiometric battlefield weather sensor, we are investigating a neural network simulation to invert radiometric measurements of atmospheric microwave radiance to obtain vertical profiles of atmospheric temperature. Given a vertical profile of atmospheric temperature and moisture, it is relatively straightforward to compute the resulting microwave radiance spectrum at the surface. The inverse problem, that of computing the temperature and moisture profiles from measurements of microwave radiance presents more problems, principally because the mathematical problem is ill-posed. The training method being adopted for our simulation is intended to exploit the advantages and ameliorate the disadvantages of this situation.

---

1. Introduction

The realistic battlefield presents severe challenges to computer systems performance. Data rates may be very large, and a crucial task is to filter significant information from irrelevant data. Rapidly changing conditions and novel threats require adaptability that is very difficult to achieve with conventional computer programs. Many tasks to be performed are of the pattern recognition type, a type of problem for which computers have so far displayed little aptitude. Information obtained is often incomplete or fragmentary, a situation posing severe difficulties for conventional computer programs.

A radically different type of computer, patterned after the biological information processor -- the brain -- may be better suited to battlefield computing environments. These computers, called neural networks, possess several characteristics which appear to make them natural candidates for information processing and knowledge representation in such an environment.

It is intrinsic to neural networks that they are parallel distributed processors, carrying out many computations simultaneously, with the computation distributed to many nodes (the neurons). Only parallel processing systems are likely to possess the computational power to solve very complex problems in real time. Another advantage of the neural network is that most neural networks have a natural, human like, method of learning (learning by example.) Neural networks can be built with a capability for generalization and exhibit a natural tolerance for incomplete data that is hard to incorporate in conventional computational schemes.

2. The idea of the Neural Network

The organization and structure of biological computers (e.g. brains) is radically different than that of the modern digital computer. Despite the fact that the switching speeds of their neuronal components are hundreds of thousands or millions of times slower than the switching speeds of transistors, biological computers are remarkably fast and powerful at some tasks, for example pattern recognition. Human and animal brains have other surprising advantages over conventional digital computers, including flexibility, tolerance of error and ambiguity, associative memory, and a natural learning mechanism.

Many of these advantages, and others, are believed to derive from the remarkable architecture of these biological computers, an extremely highly parallel, densely interconnected system of individual processors called neurons. The human brain, for example, is believed to encompass about 10 billion neurons and 100 trillion synapses, or interconnections (Rumelhart and McClelland, 1986).

Recognition of this fact, together with recognition of the fact that the conventional single processor digital computer is approaching the limits of its performance has led to an increasing interest in computer designs which emulate aspects of these biological computers, the so-called neural networks.

The term neural network has been applied to a large variety of computational architectures, but all have in common the notion of simulated neurons and synapses. The implementations of the simulated neurons and synapses vary widely, being more or less faithful to what is known about their biological antecedents (which are themselves rather various, depending on their functional specialization). A (simulated) neuron is generally considered an entity with multiple inputs and a single (usually subsequently branching) output. The neuron's output is a function of its current state, its current inputs, and (possibly) its past state history. The output of a neuron is connected to the inputs of other neurons by synapses, each of which has a weight (or multiplying factor) associated with it. The values of the weights, in conjunction with the initial state of the system, determine the computation performed by the network.

Figure 1, which is adapted from the descriptions of Rumelhart et al. (Rumelhart, Hinton, and McClelland, 1986) shows a model neuron (labelled j) with three inputs i_1, i_2, i_3 and corresponding synapses of weights w_{j1}, w_{j2}, w_{j3}. The current state $a_j(t)$ is considered to be some function of

the previous state $a_j(t-1)$ and of the weighted sum of the inputs $\sum_k w_{jk}$. The output $o_j(t)$ is a function of $a_j(t)$.

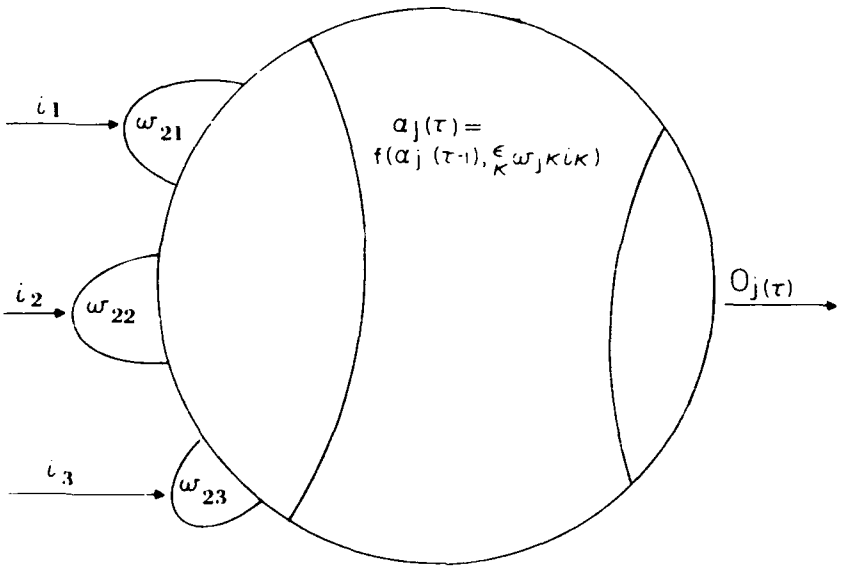


FIG. 1 A MODEL NEURON

Figure 2 shows a simple two-layer neural network with 3 external inputs, 6 neurons, 9 internal synapses, and 3 external outputs. The example used in this paper is a 5x5 version of this network. Consider the simple linear case when $o_j(t) = a_j(t) = \sum_k w_{jk} i_k$. For this case (an example of a linear perceptron) the nonzero w_{jk} form a 3x3 matrix $\underline{\mathbf{W}}$, and the output vector $\mathbf{o}$ is given by

$$\mathbf{o} = \underline{\mathbf{W}}\,\mathbf{i} \qquad (1)$$

Thus the neural network transforms the inputs into the outputs, and for this linear transformation, the weights are the elements of the transformation matrix. More general neural networks are possible, since the state may be (and for biological systems, is) a nonlinear function of the weighted inputs. Other generalizations include replacing the weighted sum of the inputs with a more general function of the weighted inputs. In our current work we have restricted ourselves to the linear case, except that we have not required the weight matrix to be square (number of inputs and outputs may be different).

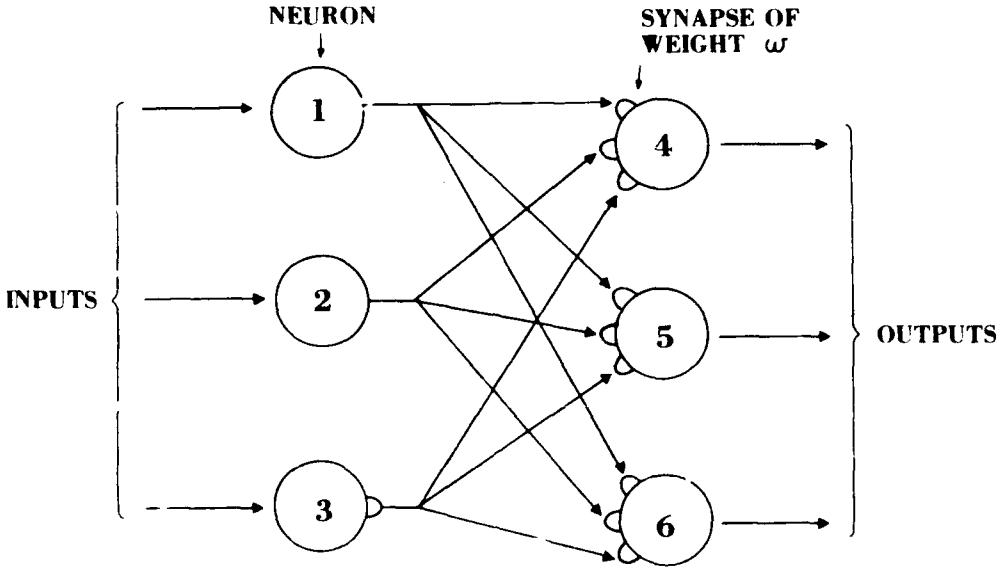


FIG. 2 A SIMPLE NEURAL NET

Since values of the weights determine the transformation of the input vector, specification of the weights is equivalent to programming the neural network. One of the reasons neural networks are interesting is that it is possible to arrange for them to learn by experience, that is, to adjust their weights based on inputs and outputs. One such learning mechanism, the one applied in this paper, is Widrow-Hoff or delta-rule learning (Rumelhart, Hinton, and McClelland, 1986) .

Delta-rule learning requires a training set, that is, a set of inputs together with the corresponding outputs. Initially, the weights are set to small random values. Inputs are supplied and the corresponding outputs obtained, compared with the desired outputs, and an error signal generated. This error signal is used to generate corrections to weights as follows:

$$\Delta w_{ij} = \eta \, (t_i(t) - o_i(t)) \, i_j(t) \qquad (2)$$

where Δw_{ij} is the change in weight w_{ij}, η is a learning rate factor, t_i is the target output from the ith unit, o_i is the actual output from the ith unit, and i_j is the jth input.

Much of the work with neural networks has focussed on simulation of brain function, associative memory, and pattern recognition. For reasons to be discussed in the following, we believe that neural network techniques may also have advantages for certain types of computational tasks that are usually addressed by more conventional computational techniques.

3. Retrieval of Temperature Profiles from Radiometric Measurements

Ground-based measurements of atmospheric brightness temperature at several microwave frequencies permit some inference about the vertical temperature structure of the atmosphere (Barber et al., 1986). In the 20 GHz to 60 GHz region, the measured brightness temperatures satisfy (to a good approximation) the following equation:

$$T_{b\nu} = \int_0^\infty T(s)\, \alpha_\nu(s)\, \exp[-\int \alpha_\nu(s')ds']\, ds + T_{b\nu}^\infty \exp[-\int_0^\infty \alpha_\nu(s)ds] \quad (3)$$

where $T_{b\nu}$ = the downwelling microwave brightness temperature at frequency ν

$T(s)$ = temperature at height s

$\alpha_\nu(s)$ = the absorption coefficient

and

$T_{b\nu}^\infty$ = the downwelling cosmic microwave background brightness temperature above the atmosphere.

Inferring atmospheric temperature structure from microwave brightness temperature measurements thus becomes the problem of solving equation 3 to find T(s). Problems of this type are often referred to as inverse problems since it is relatively simple to compute $T_{b\nu}$ from a knowledge of T(s) but more difficult to obtain T(s) from $T_{b\nu}$. While there are many techniques for attacking such problems (Westwater and Sweezy, 1983), each has limitations. Our intent in this paper is to investigate the applicability of a technique based on the idea of the neural network to this problem.

4. Defining a Neural Network for the Inversion Problem

Equation 3 has a great deal in common with the linear transformation problems of equation 2, although the transformation involved is not in general linear. In discrete approximation, the right hand side of equation three takes a set of temperatures defined at discrete heights and produces from them the corresponding microwave brightness temperatures at discrete frequencies. While this direct computation is fairly straightforward, various obstacles exist for the inverse computation (Twomey, 1977).

Our objective is to perform the inverse calculation with a neural network simulation. Inputs to the neural network will be measured microwave brightness temperatures and other relevant measurements, for example, surface measurements of temperature and pressure. Outputs will be estimated temperature at various heights.

As mentioned above, our intention is to train this neural network using the delta-rule training scheme. This requires a teaching set of inputs and their corresponding outputs. Because it is straightforward to do the direct computation, there is a natural way to get such a teaching set. In our case, we have assembled a large set of radiosonde (meteorological balloon) observations of atmospheric temperature and humidity profiles. From each such profile, equation 3 permits computation of a vector of corresponding microwave brightness temperatures. This vector of brightness temperatures (augmented by surface observations or other data) can then serve as the input vector **i** and the radiosonde temperature observations as the training outputs **o**. This neural network is indicated schematically in figure 3.

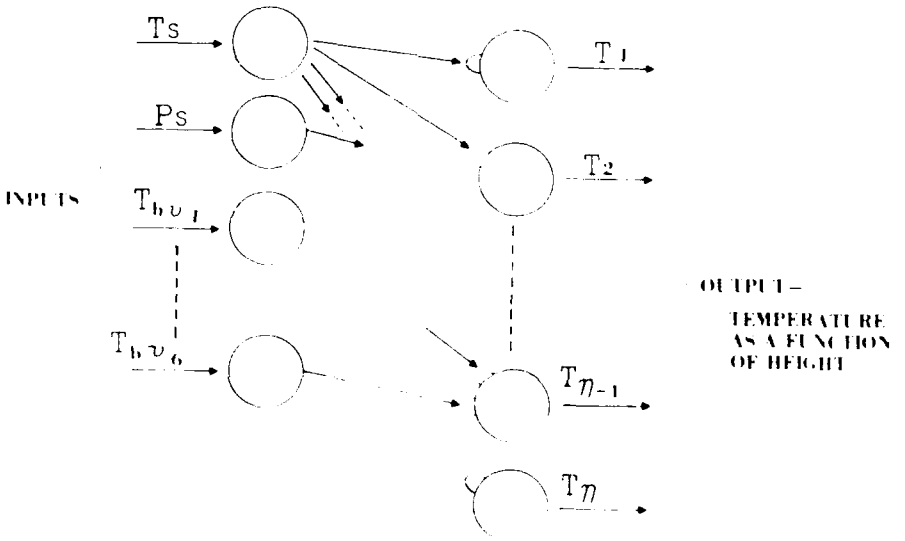


FIG. 3 A SIMPLE NEURAL NETWORK FOR RETREVING TEMPERATURE PROFILES

5. Simulations and Current Results

Implementation of the system described above is currently in progress. Results to date consist of simulation of slightly simpler linear systems of the form of equation 1. For each of these examples, a matrix $\underline{\mathbf{A}}$ was chosen, and a set of psuedo-random vectors generated (by using a random number generator to generate their components). Next, for each random vector **r**, the corresponding vector $\mathbf{v} = \underline{\mathbf{A}}\,\mathbf{r}$ was computed, and the resulting pairs of vectors were stored in a file. Since our objective was to devise a neural network which computed $\mathbf{r} = \underline{\mathbf{W}}\,\mathbf{v}$, $\underline{\mathbf{W}} = \underline{\mathbf{A}}^{-1}$ is an analytic solution, which facilitates comparison. Thus, in our examples, we were merely attempting to invert matrices. Note that we are not advocating neural networks for

solution of linear systems, but they do provide convenient and well behaved examples.

The vectors **r** served as our input vectors, the corresponding vectors v as our training set. The training rule actually used was a modification of the delta-rule discussed above, with the weight change for each iteration given by

$$\Delta w_{ij}(t) = \eta\ (t_i(t) - o_i(t))\ i_j(t) + \alpha\ \Delta_{ij}(t-1) \quad (4)$$

where α is a so-called momentum factor and $\Delta_{ij}(t-1)$ is the weight adjustment in the previous iteration and parenthetical t's refer to the current iteration (time). Addition of this factor improves convergence (Rumelhart, Hinton, and Williams, 1986).

The following tables illustrate the progress of a sample training session for a 5x5 neural network. Table 1 shows our example matrix $\underline{\mathbf{A}}$. Table 2 shows the inverse matrix $\underline{\mathbf{A}}^{-1}$ as obtained by conventional linear algebra. Subsequent tables show the matrix $\underline{\mathbf{W}}$ after 100 iterations, 1001 iterations, 2001 iterations, 5001 iterations, and 10001 iterations. In each case, 1000 pairs of vectors were generated. Where more iterations were needed for convergence, the same sets were repeatedly run through the neural net simulation.

TABLE 1.

Original matrix $\underline{\mathbf{A}}$

3.0	1.0	0.0	2.0	0.0
3.0	0.0	2.0	1.0	1.0
0.0	2.0	2.0	1.0	3.0
1.0	2.0	3.0	2.0	2.0
2.0	3.0	1.0	3.0	0.0

TABLE 2.

Inverted Matrix $\underline{\mathbf{A}}^{-1}$

-.348	.652	.304	-.783	.435
-1.174	.826	.652	-1.391	1.217
-.609	.391	-.217	.130	.261
1.609	-1.391	-.783	1.870	-1.261
.652	-.348	.304	217	-.565

TABLE 3.

$\underline{W}$ After 100 Iterations.				
.249	.190	.059	-.040	-.008
.019	-.095	.170	.085	.336
-.566	.345	-.195	.139	.234
.290	-.367	-.199	.188	-.274
.349	-.112	.443	-.174	-.337

TABLE 4.

$\underline{W}$ After 1001 Iterations				
-.046	.431	.180	-.443	.176
-.573	.385	.405	-.716	.703
-.595	.381	-.223	.146	.249
.932	-.895	-.505	1.109	-.682
.496	-.233	.369	.041	-.431

TABLE 5.

$\underline{W}$ After 2001 Iterations				
-.208	.550	.247	-.625	.315
-.896	.622	.538	-1.079	.979
-.602	.387	-.220	.137	.256
1.295	-1.161	-.654	1.517	-.993
.580	-.295	.334	.136	-.503

TABLE 6.

$\underline{W}$ After 5001 Iterations				
-.334	.642	.299	-.767	.423
-1.146	.806	.641	-1.360	1.194
-.608	.391	-.218	.131	.260
1.578	-1.368	-.770	1.835	-1.234
.645	-.343	.307	.209	.559

TABLE 7.

$\underline{W}$ After 10001 Iterations				
-.348	.652	.304	-.782	.435
-1.173	.826	.652	-1.391	1.217
-.609	.391	-.217	.130	.261
1.608	-1.391	-.782	1.869	-1.260
.652	-.348	.304	.217	-.565

6. Conclusions and Future Plans

Although the process of convergence is quite slow, with the weight matrix showing little resemblance to the inverse matrix even after 1001 iterations, $\underline{W}$ does gradually converge to $\underline{A}^{-1}$. Rate of convergence is affected strongly by the size of the parameters α and η, and by the choice of training vectors.

Because of the simplicity of the cases addressed to date it is not yet feasible to judge the usefullness of the method. The results for the simple linear systems are encouraging, but we have yet to address the difficulties posed by non-linear systems or even the more familiar difficulties associate with ill-conditioned linear systems.

References

(Barber et al., 1986) Barber, T. L., Young P. Yee, Edward M. Measure and D. R. Larson, "Tactical Weather Intelligence for Artillery". Proceedings of Seventh EOSAEL/TWI Conference, Las Cruces, NM, 2-4 Dec, 1986.

(Rumelhart, Hinton, and McClelland, 1986) Rumelhart, D. E., G. E. Hinton, and J. L. McClelland, "A General Framework for Parallel Distributed Processing." in Parallel Distributed Processing, David E. Rumelhart and James L. McClelland, eds., MIT Press, Cambridge, MA, 1986.

(Rumelhart, Hinton, and Williams, 1986) Rumelhart, D. E., G. E. Hinton, and R. J. Williams, "Learning Internal Representations by Error Propagation." in Parallel Distributed Processing, David E. Rumelhart and James L. McClelland, eds., MIT Press, Cambridge, MA, 1986.

(Rumelhart and McClelland, 1986) Rumelhart, D. E., and J. L. McClelland, "PDP Models and General Issues in Cognitive Science," in Parallel Distributed Processing, David E. Rumelhart and James L. McClelland, eds., MIT Press, Cambridge, MA, 1986.

(Westwater, 1983) Westwater, E. R., and W. B. Sweezy, "Profile retrieval algorithms used in thermal sounding of the atmosphere." Proceedings of SPIE Conference on Inverse optics, Arlington, VA, Apr 6-8, 1983.

(Twomey, 1977) Twomey, S., Introduction to the Mathematics of Inversion in Remote Sensing and Indirect Measurements, pp.. Elsevier, New York, 1977.

Decision Support System Software for the Battlefield Environment[1]

S. A. Barrett, S. W. Barth, and K. H. Gates
PAR Government Systems Corporation
New Hartford, NY 13413, USA

ABSTRACT

PAR Government Systems Corporation AI Group has developed generalized decision support system (DSS) software to aid in building decision aids for battle management. The modular DSS software is comprised of a blackboard system framework coupled with an expert system framework, augmented by packages for menu-driven windowing and map feature presentation. The software has been used to build a decision aid to support a Command, Control, and Communications Order of Battle (C^3OB) analyst in identifying fixed and mobile targets on the battlefield.

1. INTRODUCTION

Decision support systems for battle management often require manipulation of a wide range of modelled entities and application of diverse problem-solving methods. Additionally, the decision-making process which is supported by the system may evolve or change dramatically as new approaches to the problem are formulated. Decision support system (DSS) software which is modular in design and may be flexibly configured can reduce the amount of effort needed to generate new decision support systems as well as to modify existing decision support systems over their lifetimes.

PAR Government Systems Corporation AI Group has developed generalized decision support system software under contract to Rome Air Development Center. The software was developed to facilitate the creation and modification of decision aid prototypes to support Command, Control, and Communications Order of Battle (C^3OB) analysts in

---

[1]This work was supported in part by Rome Air Development Center contract F30602-C-87-0112.

identifying targets for which C^3 countermeasures (C^3CM) may be effectively employed. The software is comprised of a framework for building blackboard systems, a rule-based expert system framework, a menu-driven windowing package, and a map feature presentation package. The software was developed with currently available hardware as the intended delivery platform.

This paper describes the generalized decision support system software and an application that is currently under development. Section 2 provides a high level description of the hardware and software environment for which the system software was developed. Section 3 describes each of the major software components of the DSS software. Section 4 describes an application of the software system to a battle management problem.

2. ENVIRONMENT

2.1 SOFTWARE ENVIRONMENT

Table 1 contains a description of the software environment employed by the generalized DSS software. Much of the software environment was selected for reasons of portability. A decision support application (or decision aid) may be developed (with the exception of the graphical interface) on one of many other UNIX-based systems, due to the portability of the expert system and blackboard system frameworks, which are written in the C and C++ programming languages. The selection of the XENIX operating system was motivated by our contract's intended delivery environment.

TABLE 1. GENERALIZED DSS SOFTWARE ENVIRONMENT

C programming language
C++ programming language
Object-Oriented Program Support library
XENIX V (2.3) operating system

The C++ programming language (Stroustrup, 1986) was chosen in order to take advantage of object-oriented programming techniques for creating a modular and extensible blackboard system framework. C++ is a superset of the C language which supports object-oriented programming (among other things), and is implemented as a translator which emits C source code. Thus, it retains most of the portability of the C language. The Object-Oriented Program Support (OOPS) library, developed at the National Institute of Health (Gorlen, 1986), provides a set of C++ classes

from which many of the blackboard system framework components are built.

2.2 HARDWARE ENVIRONMENT

Table 2 contains a description of the hardware (delivery) environment for the generalized DSS software. The selection of the hardware was motivated by its similarity to that of the Intelligence Work Station (IWS) used by the 9th Tactical Intelligence Squadron (TIS).

TABLE 2. GENERALIZED DSS HARDWARE ENVIRONMENT

IBM PC/AT with Intel 80386 processor
13" color console with EGA card
19" color monitor with 8-bit IMAGRAPH board
3-button mouse

The 13 inch color console with extended graphics capability is used by the menu-driven windowing software. The 19 inch color monitor is used by the map feature presentation package. Although a mouse with three buttons is indicated in the table, only one button is needed for the current software.

3. DECISION SUPPORT SYSTEM SOFTWARE COMPONENTS

The decision support system software is comprised of four components: a blackboard system framework, an expert system framework, a menu-driven windowing package, and a map feature presentation package.

3.1 BLACKBOARD SYSTEM FRAMEWORK

The Blackboard-Building Tool for Hierarchical Inferencing Gadgets (BBTHING), is a library of building blocks which may be used to implement systems based upon the blackboard model of problem solving (Nii, 1986a; and Nii, 1986b). The library provides classes of blackboard system components which may be specialized or used directly in configuring a blackboard system.

3.1.1 The Blackboard Model of Problem Solving

The blackboard model of problem solving is one in which all information about the problem solution is kept on a hierarchically organized global data structure called the blackboard. The knowledge about how to solve the particular problem is split into logically

independent knowledge sources which respond to changes occurring on the blackboard. There is no rigid control structure which directs the problem-solving activity. Instead, knowledge is applied to the problem opportunistically. Problem solutions are produced incrementally via changes to the blackboard made by the knowledge sources.

The structure of a blackboard for a given system is entirely dependent upon the concepts or entities which describe the solution space. The blackboard is usually divided into a hierarchy of levels of analysis appropriate for the intended problem-solving activity. Each level of the blackboard is populated by objects which are germane to that particular level of analysis. Relationships between objects are represented by links.

Changes to the blackboard may only be made by knowledge sources, which contain pieces of domain knowledge needed to create a solution to the problem. This knowledge may be in the form of procedures or rules. Each knowledge source may transform information on one level of the blackboard into information on other blackboard levels.

Knowledge sources operate independently and opportunistically. All interaction between knowledge sources takes place via explicit changes to information on the blackboard. Each knowledge source keeps track of the conditions under which its knowledge may be applicable to the problem-solving process. When those conditions arise, the knowledge source tries to apply its knowledge to a portion of the current partial solution.

3.1.2 The Blackboard Framework (BBTHING)

Though different blackboard systems tend to vary widely in their configurations, their components tend to be similar in functionality. Several attempts have been made to provide generalized blackboard system components (Nii and Aiello, 1979; and Hayes-Roth 1985). BBTHING is just such a package. Object-oriented programming techniques were employed to design a set of building blocks which could be easily extended and configured to implement systems using a blackboard architecture.

BBTHING is a library of C++ classes representing typical components of blackboard systems. These classes may be used as they are or as a starting point for designing new component classes. New component classes which are derived from a class in the library inherit the behavior of that class of component (unless overridden by the designer). This gives the blackboard designer some leverage; a little programming effort results in a lot of functionality. Table 3 shows a hierarchy of classes which are

available to the blackboard system designer. Each of the groups of classes will be described briefly below.

TABLE 3. BBTHING CLASS HIERARCHY

Blackboard
Level
Node
Event
Clock Event
Initialization Event
Periodic Event
Event List
Clock Event List
Event Selector
LIFO Selector
FIFO Selector
Clock Selector
Event Strategy
Do One Strategy
Do All Strategy
Do Due Strategy
Knowledge Source
ERS Knowledge Source
Activity
Focus
Focus List
Focus Queue
Focus Stack

The Blackboard class is the class whose elements are configured to create a blackboard system application. The blackboard class coordinates the activities of the various levels, event lists and focus lists which are in the configured system.

The Level class implements a level in the blackboard hierarchy. Its primary function is to hold node objects of different types and to act as a data base which responds to queries about those nodes. Node storage and retrieval techniques may be customized by deriving a new class from the Level class.

The Node class provides the basic functionality of an object on the blackboard. Each blackboard system will typically have many classes of objects derived from the Node class, depending upon the model of the problem domain.

The Event classes provide the mechanism for signalling changes to the blackboard which "trigger" knowledge sources into action. Each blackboard system will have its own set of event types, depending upon the approach to the decision-making process which has been adopted and the knowledge sources that have been created to implement that approach. BBTHING provides a basic Event class, from which most domain-specific events will be derived. It also provides a Clock Event class for events which must occur at a specific time. Periodic Events are those events which are to recur at a given interval of time. An Initialization event is automatically generated by the Blackboard to initiate processing.

The Event List classes contain the Event objects which signal changes to the blackboard data, and coordinate the processing of these events. The Clock Event List class contains all Clock Event and Periodic Event objects. Each event list uses an Event Strategy object to determine whether some of its events should be processed next. If the current event strategy indicates that an event should be processed, the event list uses an Event Selector object to select the appropriate event to process. Event processing consists of finding Knowledge Source objects which are "triggered" by the event, and asking the blackboard to add and Activity object to its agenda.

The Event Strategy classes provide different strategies for processing events in a particular event list. A Do One Strategy object only lets an event list process a single applicable event before relinquishing control to the next event list. Applicability is determined based upon the blackboard's current Focus object. A Do All Strategy object allows the event list to process all applicable events before relinquishing control. The Do Due Strategy class is used by Clock Event Lists to determine whether any Clock or Periodic Events require attention. New strategies for processing events may be created by deriving a new class of event strategy from the Event Strategy class.

The Event Selector classes provide different styles of event management. The LIFO Selector class makes an event list operate as a stack, while the FIFO Selector class makes an event list operate as a queue. A Clock Selector object is used by Clock Event Lists to select those Clock and Periodic Events which are due for processing. Additional styles of event management may be implemented by deriving a new class of event selector from the Event Selector class.

The Knowledge Source classes provide the basic functionality for knowledge sources within a blackboard system. Different methods of solving subproblems on the blackboard may be generated by deriving new

knowledge source classes from the Knowledge Source class. The ERS Knowledge Source class provides a linkage to the Embedded Rule-Based System, allowing a rule base to function as a knowledge source.

The Activity class is used to create objects which describe triggered knowledge sources, indicating places on the blackboard where the expertise of the knowledge source is to be applied.

The Focus class is used to create objects which narrow the focus of processing to either certain types of events, certain types of nodes, or a particular knowledge source. The focus objects may be generated by a knowledge source and added to the blackboard's Focus List object.

The Focus List classes implement strategies of focussing the problem-solving behavior of the blackboard. The Focus Queue and Focus Stack classes implement FIFO and LIFO focussing, respectively. New styles of focus management may be implemented by deriving a new class from the Focus List class.

3.2 EXPERT SYSTEM FRAMEWORK

The expert system framework used in the decision support system software is the Embedded Rule-Based System (ERS) (Barth and Quinn-Jacobs, 1986). ERS is a framework for developing rule-based applications which uses a probabilistic inferencing mechanism similar to that of PROSPECTOR (Duda, Hart, and Gaschnig, 1979). ERS may be used to create consultation systems, which query the user for data or beliefs, or embedded systems, which access and modify external data during inferencing. The framework is written in the C programming language and is available for a variety of operating systems.

The ERS framework consists of an inference engine and rule base parser. Rules are written in the ERS rule base language using any text editor. (ERS rule base language templates are available for the GNU Emacs text editor, which simplifies rule base generation considerably.) During run-time initialization, the rule base file is parsed and compiled into an inference network data structure. The inference engine uses this data structure to drive the decision-making process.

The inferencing process in ERS may involve gathering evidence from the user, as is usually done in expert consultation systems, or from a set of application-dependent functions which interact with existing data sources, or both. As sufficient evidence is gathered, conclusions or advice may be reported to the user as statements of degree of belief for the top-level goal hypotheses that were defined in the rule base. Actions my be executed to

change data, or some device, based on the certainty of these hypotheses. The system continues gathering evidence, taking actions, and/or reporting advice, until no more evidence remains to be gathered, or until the user issues a quit command.

Knowledge engineering to build a rule-based system involves an iterative and incremental development process (Hayes-Roth, Waterman and Lenat, 1983). In using ERS, this process consists of a development cycle which can be broken down into several phases.

First, expert knowledge must be obtained from domain specialists, or other sources of expertise, and represented in the ERS rule base language. If the application is to be embedded in other software, some rules may require actions to be taken on a data base or device, and others may require input from data bases or devices as evidence. Functions that perform actions and evidence tests must be coded and tested.

As rules are developed, performance of the system can be evaluated by running ERS with the partially-completed rule base. Through evaluation of system performance and the explanation and debugging facilities provided by ERS, problems with the rules can be identified. These problems can be corrected by editing the rule base file to modify existing rules or add new ones.

The cycle of modifying the rule base and evaluating system performance then continues, with resulting incremental improvements to the rule base. Methods for evaluating the system's performance will, of course, depend upon the particular application of ERS, but, in general, such techniques involve comparing the system's performance to that of human experts (Adelman and Gates, 1983; Gaschnig et al., 1983).

ERS has been used for several expert system applications at PAR Government Systems Corporation, including decision aids for military unit identification and assessment of enemy intentions, analysis of data on foreign space launches, evaluation of geographic data for minefield site location, and detection of changes in digital imagery to determine air order of battle. In general, ERS can be used for any kind of application for which a probabilistic, goal-driven, backward-chaining inference mechanism is appropriate. ERS seems best suited for applications involving some kind of diagnostic task, inferring underlying causes or states from observed evidence of symptoms, with rules that involve uncertainty.

3.3 MENU-DRIVEN WINDOWING PACKAGE

The menu-driven windowing package was developed for the current hardware and operating system environments. The package is a partial implementation of Microsoft Windows which runs under the XENIX operating system. The package includes the usual complement of windowing facilities such as overlapping windows, drop-down menus, scrolling text fields, control panels, check boxes, activation buttons, and mouse-sensitive icons. The package operates in color using the EGA option of the IBM PC/AT and is operated by using the mouse as a pointing device.

3.4 MAP FEATURE PRESENTATION PACKAGE

The map feature presentation package is used to display World Data Base II map information on a high-resolution color monitor. While the package is relatively unsophisticated, it does include such capabilities as software zooming and panning, selectable display of feature types, labels on major features (such as cities), display of icons, and point to point distance measurements. The package is also capable of overlaying the graphics with a labeled grid of latitude-longitude lines.

4. AN APPLICATION OF DECISION SUPPORT SYSTEM SOFTWARE

Under our current contract for RADC, the generalized decision support system software has been used to implement an advanced development prototype decision aid supporting the 9th Tactical Intelligence Squadron within the Tactical Air Control Center (TACC) of the U.S. Central Command Air Forces (USCENTAF). The decision aid is designed to assist C^3OB analysts by providing a structured approach to the identification of targets for which C^3 countermeasures (C^3CM) may be effectively employed.

The decision aid consists of a pair of subsystems, Identification of Command and Control Operations Nodes (ICON) and Threat Event Analysis Methodology (TEAM). Figure 1 shows a very high level view of the architecture of the ICON/TEAM decision aid.

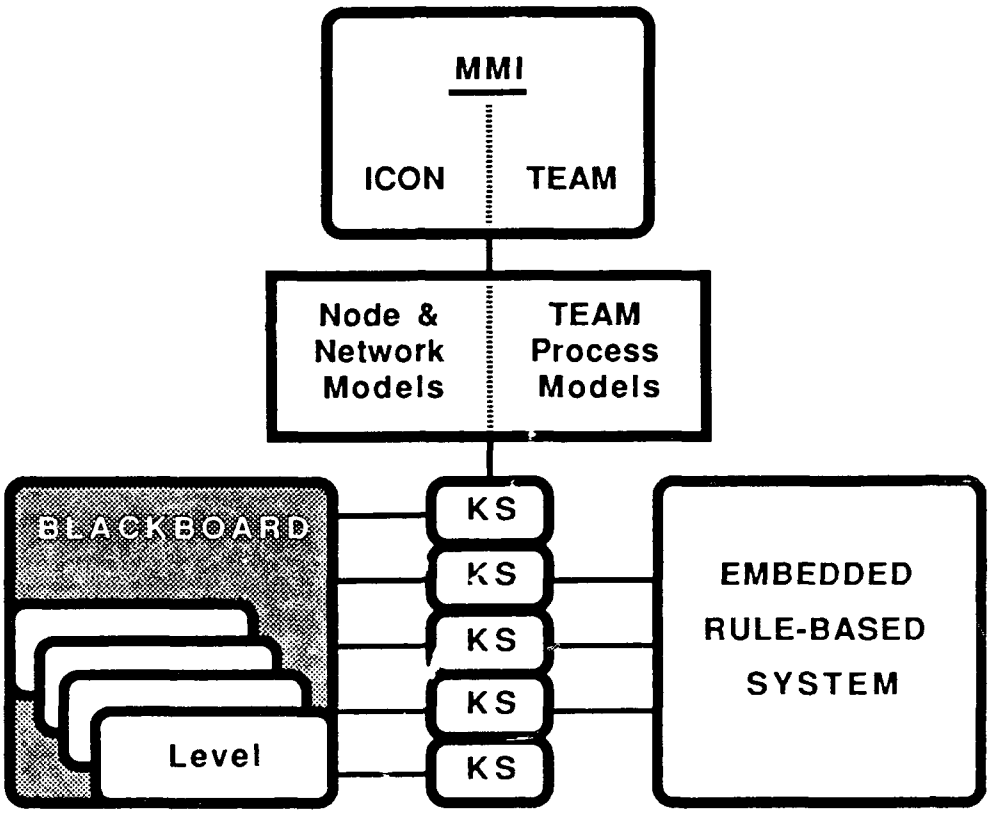


FIGURE 1. High-level ICON/TEAM architecture. The ICON subsystem consists of the blackboard, knowledge sources (labelled KS) and the Embedded Rule-Based System. The TEAM subsystem is subsumed by the TEAM half of the MMI module.

4.1 ICON

The ICON subsystem uses object-oriented programming techniques to model C^3 nodes and their organizational structure. ICON is implemented as a blackboard system with a combination of rule-based and algorithmic knowledge sources. The rule-based knowledge sources use ERS to provide the required inferencing capability. The models of the nodes and organizational structure are used by the knowledge sources and ERS to identify C^3 nodes from incoming correlated intelligence data.

The initial prototype of the ICON subsystem includes 6 C^3 nodes and their subordinate hierarchies. Table 4 lists the nodes to be identified by the initial ICON prototype.

TABLE 4. INITIAL ICON PROTOTYPE C^3 NODES

Combat Control Center (CCC)
Combat Control Group (CCG)
Vectoring and Target Designation Point (VTDP)
SA-4 Brigade Command Post
SA-6 Regimental Command Post
SA-8 Regimental Command Post

4.2 TEAM

The TEAM subsystem is a partial implementation of the TEAM concept. The TEAM concept, sponsored by Tactical Air Command Headquarters, is a methodology for assessing C^3 target values, evaluating the effect of friendly electronic combat capabilities, and determining the best application of available assets.

The TEAM subsystem consists of object-oriented models of C^3 structures and events for various tactical situations. For example, a typical SAM engagement sequence would be represented as a TEAM process model. The descriptions of the events make use of the Node and Network models that are a part of the ICON subsystem.

The MMI module lets the user review the TEAM process, which are depicted graphically, to support analysis of the current situation in the ICON/TEAM decision aid.

5. SUMMARY

We have described a generalized package of decision support system (DSS) software to aid in building decision aids for battle management. The DSS software, comprised of a blackboard system framework coupled with an expert system framework, has been used to build a decision aid to support a Command, Control, and Communications Order of Battle (C^3OB) analyst's task of identifying fixed and mobile targets on the battlefield. The motivation for developing such a package was the requirement for flexibility in configuring decision support systems, due to the changes which may occur in the approach to solving battle management problems. The generalized DSS software package is modular in design and may be flexibly configured to reduce the amount of effort needed to generate new decision support systems as well as to modify existing decision support systems over their lifetimes.

ACKNOWLEDGMENTS

The authors would like to thank Knowledge Systems Concepts, Inc., who developed the TEAM process models and the initial menu-driven windowing and map feature presentation packages under subcontract to PAR Government Systems Corporation. We would also like to thank Sean Lanigan for his tireless work in improving these packages and developing the ICON user interface.

REFERENCES

Adelman, L., and K. H. Gates, 1983: Test Analysis Report for the Duplex Army Radio/Radar Targeting Aid (DART), PAR Report #83-84, PAR Technology Corporation, PAR Technology Park, 220 Seneca Turnpike, New Hartford, NY 13413.

Barth, S. W., and D. P. Quinn-Jacobs, 1986: Embedded Rule-Based System (ERS) User's Manual, PGSC Report #86-64, PAR Government Systems Corporation, PAR Technology Park, 220 Seneca Turnpike, New Hartford, NY 13413.

Duda, R. O., P. E. Hart, and J. Gaschnig, 1979: Model Design in the PROSPECTOR Consultant System for Mineral Exploration. In D. Michie (Ed.), Expert Systems in the Micro-electronic Age, Edinburgh University Press, Edinburgh.

Gaschnig, J., P. Klahr, H. Pople, E. Shortliffe, and A. Terry, 1983: Evaluation of Expert Systems: Issues and Case Studies. In F. Hayes-Roth, D. Waterman, and D. Lenat (Eds.), Building Expert Systems, Addison-Wesley Publishing Company, Reading, Massachusetts.

Gorlen, K. E., 1986: Object-Oriented Programming Support Reference Manual, Computer Systems Laboratory, Division of Computer Research and Technology, National Institute of Health, Bethesda, Maryland 20892.

Hayes-Roth, B., 1985: A Blackboard Architecture for Control, Artificial Intelligence, Vol. 26 No. 2, 251-321.

Hayes-Roth, F., D. A. Waterman and D. B. Lenat (Eds.), 1983: Building Expert Systems, Addison-Wesley Publishing Company, Reading, Massachusetts.

Nii, H. P., and N. Aiello, 1979: AGE (Attempt to Generalize): A Knowledge-Based Program for Building Knowledge-Based Programs, IJCAI 6, 645-655.

Nii, H. P., 1986a: Blackboard Systems: The Blackboard Model of Problem Solving and the Evolution of Blackboard Architectures, AI Magazine, Vol. 7 No. 2, 38-53.

Nii, H. P., 1986b: Blackboard Systems: Blackboard Application Systems, Blackboard Systems from a Knowledge Engineering Perspective, AI Magazine, Vol. 7 No. 3, 82-106.

Stroustrup, B., 1986: The C++ Programming Language, Addison-Wesley Publishing Company, Reading, Massachusetts.

AVENUE OF APPROACH GENERATION

D. Powell and G. Storm
Los Alamos National Laboratory
Los Alamos, NM 87545

ABSTRACT

Los Alamos National Laboratory is conducting research on developing a dynamic planning capability within an Army corps level combat simulation. Central to this research is the development of a computer based ability to "understand" terrain and how it is used in military planning. Such a capability demands data structures that adequately represent terrain features used in the planning process. These features primarily relate to attributes of mobility and visibility. Mobility concepts are abstracted to networks of mobility corridors. Notions of visibility are, for the purposes of planning, incorporated into the definition of key terrain. Prior work at Los Alamos has produced algorithms to generate mobility corridors from digitized terrain data. Mobility corridors, by definition, are the building blocks for avenues of approach, and the latter are the context in which key terrain is defined. The purpose of this paper is to describe recent work in constructing avenues of approach, characterization of avenues using summary characteristics, and their role in military planning.

1. INTRODUCTION

According to military doctrine (CGSC 1986) understanding the restrictions and opportunities offered by terrain is fundamental to the planning of a mission. The military utility of an area is viewed from the perspectives of observation and fire, cover and concealment, obstacles, key terrain, and avenues of approach and mobility corridors (OCOKA). Earlier work at Los Alamos (Powell 1987), proposed conceptual approaches to the computer analysis of digitized terrain in support of terrain analysis. A companion paper (Powell et al. 1988) details work in computer based terrain analysis based on these concepts. While other work in computer based terrain analysis exists (Welo, 1986; Diaz et al. 1986; plus a large body of work at Engineering Topographic Laboratory and Waterways Experiment Station), it primarily is for visual interpretation by a human analyst or planner. The Los Alamos effort is directed to the autonomous interpretation of the terrain data structures by a computer, to support computer based planning activities for subsequent execution within a combat simulation. This approach requires the extraction and representation of relevant terrain features as well as the modeling of basic terrain analysis skills used by military analysts.

2. TERRAIN DATA

The terrain data base used is a 97 by 125 kilometer region near the inter-German border (lower left hand reference is NA0063). The data are 100 meter resolution, each terrain point is annotated with several attributes describing surface and cultural features. Although the terrain computations are described in detail elsewhere (Powell et al. 1988), briefly, the terrain data are processed to ascertain mobility attributes: go, slow-go, and no-go as defined by doctrine (CGSC 1986). Next, no-go terrain points are aggregated into regions, the boundaries of which are processed to determine mobility corridors. Mobility corridors are relatively open areas that permit movement and maneuver for a given size unit. Mobility corridors are modeled by a network of nodes and edges. Edges maintain information on mobility within the corridor, corridor width characteristics, and measures of distance, cover, and concealment. Avenues of approach are constructed from mobility corridors that satisfy size and proximity criteria.

An avenue of approach for a given size unit consists of two or more mobility corridors for units one size smaller, e.g. a brigade avenue of approach contains two or more battalion size mobility corridors. The constituent mobility corridors must satisfy certain distance criteria, given in Table 1.

TABLE 1. AVENUE OF APPROACH SIZE

Avenue of Approach	Mobility Corridor	Proximity (Km)
Division	Brigade or Regiment	10
Brigade or Regiment	Battalion	6
Battalion	Company	2

3. CHARACTERIZATION OF AVENUES OF APPROACH

3.1 DATA STRUCTURES

The terrain objects are represented by the Symbolics-Lisp Flavor System; this facet of Symbolics-Lisp allows real-world entities and their characteristics to be modeled as lisp objects. Mobility corridors, for example, have characteristics such as length, width, cover, etc.; a mobility corridor can be modeled by a lisp object with those same characteristics. This is a method referred to as object oriented programming.

Since many real-world entities have characteristics in common, object oriented programming provides a convenient method for modeling the common characteristics of separate entities. A hierarchical representation of common characteristics of terrain objects is facilitated by an inheritance mechanism. For example, avenues of approach and mobility corridors are separate entities, but

they do have common terrain characteristics, such as cover, concealment, and average width. At the same time, mobility corridor objects will have characteristics, such as sidelines, that avenue of approach objects will not have, and avenue of approach objects will have characteristics that mobility corridors will not have. Therefore, a class of general terrain characteristics can be defined so that both avenue of approach objects and mobility corridor objects will inherit their common characteristics from the same source. The values of the inherited characteristics will be unique to the specific object that they are associated with. This hierarchical inheritance mechanism accurately models the similarities of real-world entities.

This program models three main classes of objects: edges, mobility corridors, and avenues of approach. Each of these objects contains terrain characteristics combined with individual characteristics. Fig. 1 shows the relationship between these objects with their unique characteristics and the common characteristics that they each inherit.

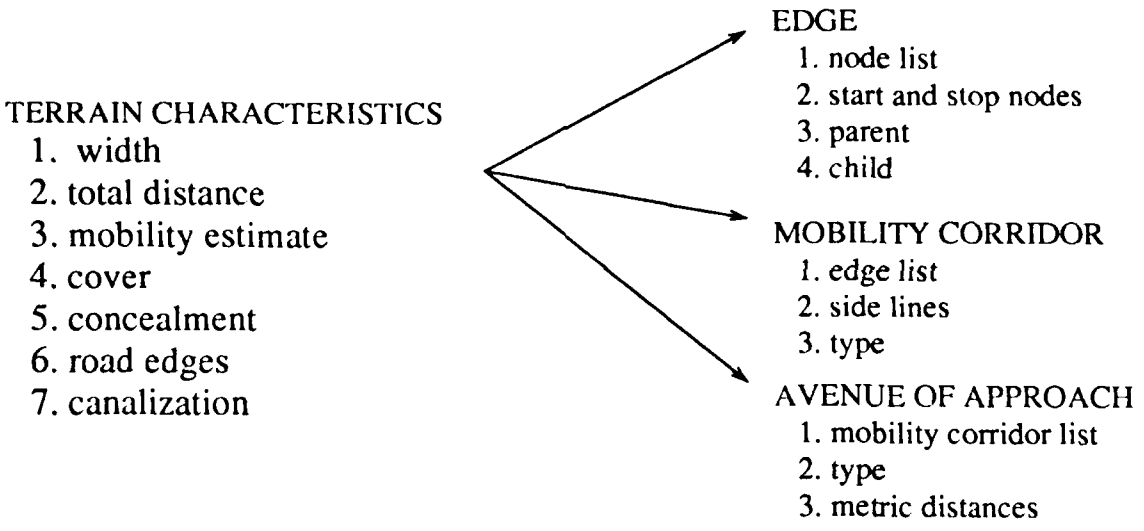


Figure 1. Inheritance of terrain characteristics by edges, mobility corridors, and avenues of approach.

Edge objects are the fundamental building blocks in this method of avenue of approach generation. Each instance of an edge object contains terrain characteristics as well as node coordinates used to map the edge. In addition, since edge instances are linked together to form mobility corridor instances, parent and child fields are used to track the edge instances that form a corridor.

As stated above, an instance of a mobility corridor object is made up of edge instances. The terrain characteristics of the edges are combined to give the corridor its terrain characteristics. Mobility corridor instances have an edge list, side lines, and type associated with them as well.

The avenue of approach instances are built from the mobility corridor instances. Here again it is the terrain characteristics of the mobility corridor

instances that are used to determine the terrain characteristics of the avenue of approach instance. The avenue instances have a mobility corridor list, a type, and a metric of the distance between corridors.

There are two other important objects. The area of interest designates the area of the terrain within which the search for avenues will take place. Presently the area of interest is a closed polygon. The exclusion list contains the edges that cross or intersect the boundary of the area of interest. They are used as the start and stop edges for finding mobility corridors.

3.2 COMBINE EDGES INTO MOBILITY CORRIDORS

Generation of a mobility corridor instance begins with the selection of two exclusion edges from the exclusion list. These two edges form the start and stop edges of the mobility corridor. An A* path finding algorithm (Nilsson 1980; Winston 1984) locates the minimum cost path, if it exists, with respect to a cost function based on the length of the mobility corridor. One mobility corridor instance is generated for each combination of two exclusion edges.

Once the mobility corridor instances are obtained, they are classified according to size (Table 2). Note that a mobility corridor of a particular size is technically an avenue of approach of the next smaller size; e.g. a battalion mobility corridor can be considered a company avenue of approach. Each of the edge instances used to form a mobility corridor instance has associated with it some average width. The width used to categorize a mobility corridor instance is the average of the widths of the edge instances.

TABLE 2. MOBILITY CORRIDOR WIDTH

Mobility Corridor Size	Minimum Doctrinal Width (Km)
Division	6.0
Brigade or Regiment	3.0
Battalion	1.5
Company	0.5

4. CONSTRUCTION OF AVENUES OF APPROACH

4.1 COMBINE MOBILITY CORRIDORS INTO AVENUES OF APPROACH

Mobility corridor instances of the same size are evaluated to determine if they satisfy the proximity criteria listed in Table I. If the proximity criteria are satisfied, then the two mobility corridor instances in question can be combined into an instance of an avenue of approach. A measure of the distance between the mobility corridors, or a metric, is devised and used to estimate proximity. Application of relevant metrics to mobility corridor descriptors seems to have the

highest fidelity to doctrinal terrain feature definitions. Although other conceptual approaches to evaluating proximity exist, this work investigates the derivation and application of suitable metrics.

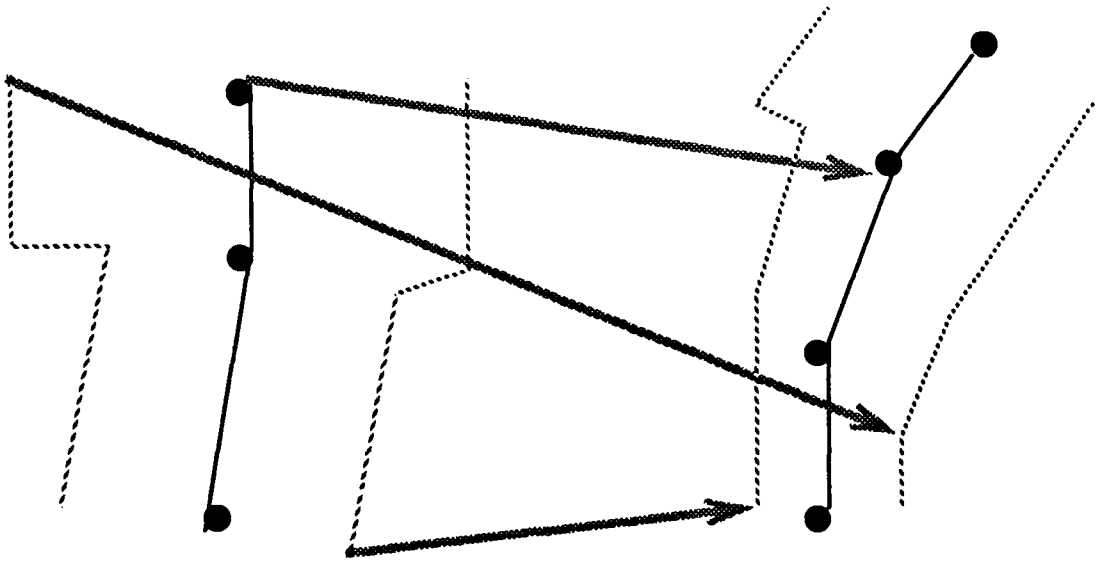
4.2 MAXMIN METRIC TO DETERMINE DISTANCE BETWEEN MOBILITY CORRIDORS

The MaxMin metric of two mobility corridor instances is calculated in the following manner: first, the minimum distance from one node of one of the mobility corridor instances, called the reference mobility corridor, to each node on another mobility corridor instance is found. This is repeated for all nodes on the reference mobility corridor instance; the maximum of these minimum distances is the value of the MaxMin Metric. Equation (1) describes the MaxMin metric where M and N are mobility corridor instances, n_{Mi} is the ith node of corridor M, n_{Nj} is the jth node of corridor N, and d(x,y) is the Euclidean distance metric.

$$\delta(M, N) = \max_i \min_j d(n_{Mi}, n_{Nj}) \quad (1)$$

By varying the line of reference that the nodes come from, there are three meaningful variations of the MaxMin metric. The first variant applies the MaxMin metric to the center lines of the mobility corridors. Each mobility corridor has associated with it a center line that splits the corridor into two halves; the center line traces the direction and length of the corridor. In addition to the center line, each mobility corridor has associated with it two side lines. The side lines are defined to follow the direction of the corridor at a distance of half the corridor width from the center line. The inner side and outer side of a mobility corridor are defined when determining proximity; the inner side is the side that is closest to the other mobility corridor being evaluated. Conversely, the outer side of a mobility corridor is the side farthest from the other mobility corridor. The second variant, the inner MaxMin, is the MaxMin metric using nodes on the inner side lines of the mobility corridors. The third variant, the outer MaxMin, uses nodes on the outer side lines.

The notion of mutual support between forces on the mobility corridors is reflected in these variants of the metric. Mutual support is the extent to which a friendly force on one mobility corridor will be able to provide direct or indirect fire to support a friendly force on another corridor. Using this military interpretation, the outer MaxMin guarantees the greatest level of mutual support. Mutual support is always possible regardless of the random placement of the friendly forces on their respective corridors. The center line MaxMin reflects an average capability to provide mutual support, the assumption is that both forces will remain on the center line of their corridor. Mutual support regarding to the inner MaxMin is only possible when both forces are on the inner sides of their corridors; this is the lowest level metric in terms of mutual support provided.



Reference Mobility Corridor Second Mobility Corridor

Figure 2. The center line, inner, and outer MaxMin distances.

Figure 2 shows the center, the inner, and the outer line MaxMin distances from the reference mobility corridor to a second mobility corridor.

4.3 CLASSIFICATION OF AVENUES OF APPROACH

Using the notions of proximity provided from the above metrics, the criteria listed in Table 1 can be used to identify and class the avenues of approach, but this classification is not always straightforward. In the case of extremely wide mobility corridors, there can exist a large difference between the proximity estimates of the MaxMin variants. A interesting situation arises when one or more of the metrics meet the criteria while the other metric(s) fail. Resolving this difference for a particular situation requires modeling the tactical judgement used by human analysts in similar situations.

While only two mobility corridors are now considered in the generation and classification of avenues of approach, additional corridors could be used. A three mobility corridor avenue would provide less restrictive mobility as well as increased mutual support. Modification of the metrics would provide a means for estimating proximity on three or more mobility corridors.

4.4 AN EXAMPLE OF AN AVENUE OF APPROACH

Figure 3 provides an example of a brigade avenue of approach generated from two mobility corridors. The solid dark lines are the edges that form the mobility corridor network through the German terrain. The dashed lines outline the area of

interest; the base is designated with longer dashes. The location where edges intersect the area of interest boundary are shown by the circles. The two mobility corridor instances are highlighted. At the right of the figure, a summary of the metrics is listed. Under the heading MaxMin Metric, the outer, center line, and inner values are shown. The avenue of approach formed, in this figure, has been classified as a brigade avenue according to the center line and inner values, because they are within the brigade criteria listed in Table 1. Note how much larger the outer value is with respect to both the center line and inner values.

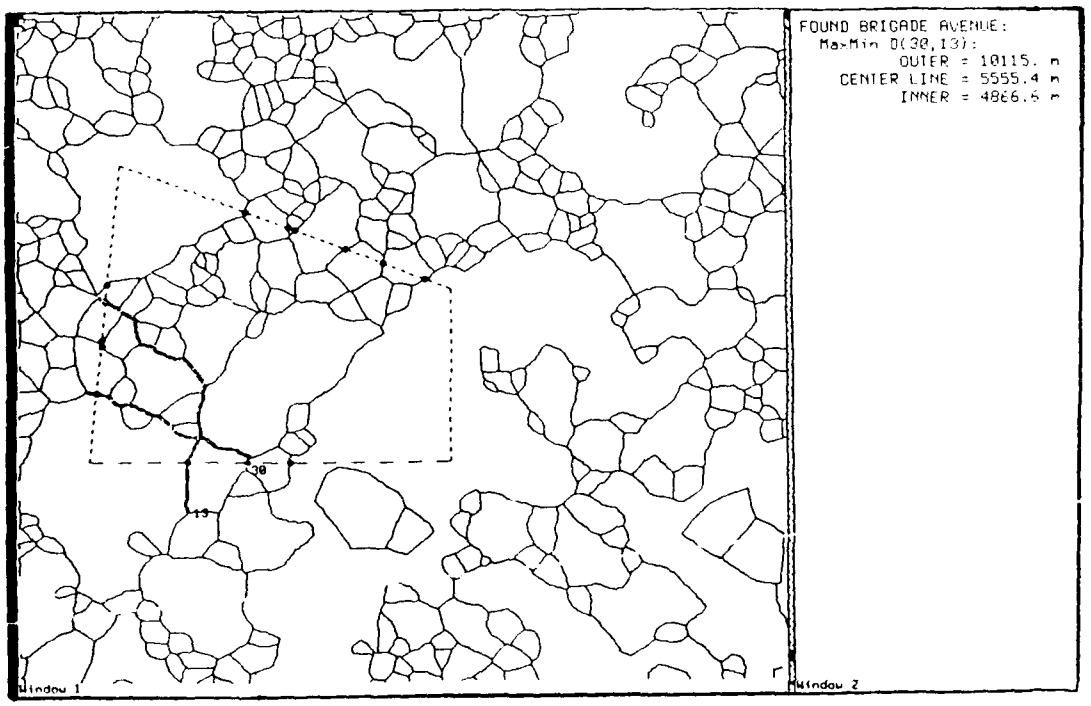


Figure 3. An example of a Brigade Avenue of Approach.

5. RELATION OF AVENUES OF APPROACH TO PLANNING

Given the avenues of approach, the planner may now focus developing courses of action. While determining courses of action, the planner must array the forces necessary to accomplish the mission. Table 3 shows the size units to be arrayed at various planning levels and the size avenue of approach utilized.

TABLE 3. PLANNING GUIDE FOR LEVEL UNIT TO BE ARRAYED

Planning Level	Avenue Size	Array Force
Corps	Division	Brigades
Division	Regiment and Brigade	Battalions
Brigade	Battalion	Companies

The initial array differs for offensive and defensive courses of action. In the offense, the process of arraying forces begins by allocating forces to all avenues of approach. The allocation is determined by the enemy units positioned to affect the friendly forces moving along the avenue. The array should give the planner an understanding of the number of forces needed to attack along all the avenues. For defense, forces are arrayed to block the enemy avenues of approach into the unit's area of operations. Here again, the planner should gain an understanding of the forces needed to provide a sound defense.

CONCLUSION

Thorough terrain analysis is essential in planning a mission. Computer based planning is being aided by representation of domain concepts and processes, e.g. human terrain analysis skills. A foundation of work aimed at deriving and representing avenues of approach generation has begun. Several metrics for determining the distance between mobility corridors have been formulated, but the work is incomplete. More study and evaluation is required to asses the validity of this approach.

REFERENCES

Diaz, A., and Smith, E., 1986: Brigade Planner: A Brigade Operational Planning Tool, Technical Document TRAC-WSMR-TD-7-86, U.S. Army TRADOC Analysis Center, White Sands Missile Range, NM.

Nilsson, N. J., 1980: Principles of Artificial Intelligence, Tioga Publishing Company, Palo Alto, CA., pp 74-76.

Powell, D., 1987: Computer Based Terrain Analysis for Operational Planning, Precedings of the 1987 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1022-1026.

Powell, D., Wright, J., Slentz, G., and Knudsen, P., 1988: A Terrain Representation for Combat Simulation, presented at U.S. Army Symposium on Artificial Intelligence Research for Exploitation of the Battlefield Environment, November 15-16, 1988.

U.S. Army Command and General Staff College, 1986: The Command Estimate, Student Text 100-9, Fort Leavenworth, Kansas 66027.

Welo, R., 1986: Mission, Enemy, Terrain, and Troops - Computer Aided Planning [METT-CAP], Technical Report TRAC-WSMR-TR-14-86, U.S. Army TRADOC Analysis Center, White Sands Missile Range, NM.

Winston, P. H., 1984: Artificial Intelligence, Addison-Wesley Publishing Company, Inc., pp. 101-113.

Representations to Support Reasoning on Terrain

D. R. Powell, J. C. Wright, G. Slentz, and P. Knudsen
Los Alamos National Laboratory
Los Alamos, New Mexico 87545, USA

ABSTRACT

Los Alamos National Laboratory has been cooperating with the Training and Doctrine Command of the US Army to develop a Corps level combat simulation for quick turn around studies. The simulation of ground combat requires representation of combat units, unit activities, command and control, and terrain. This simulation model emphasizes command and control with particular attention to the potential for automating operational planning. As terrain analysis is an essential part of Army operational planning, this has direct influence on the representation of terrain. The availability of digitized terrain makes it feasible to apply computer based techniques to emulate the terrain analysis process for use in the planning cycle. This paper describes processes used to calculate relevant terrain features for use in a simulation model.

INTRODUCTION

Combat simulations have become a major tool in analyzing military operational capabilities. In general, these simulations model the physical processes quite well, particularly in contrast to decision making processes, which are often poorly modeled. Improvement in modeling cognitive processes should start with command and control. Significant improvement of the fidelity of representation of command and control functions in combat simulations requires the development of a planning capability for the units in the simulation, producing data that is subsequently executed by the model. Currently most mission planning is done by analysts, who develop a detailed event scenario to be input into the simulation. Tools to support an automated planning system, used as a scenario builder, would greatly reduce the labor intensive process of scenario development and consequently, study turn around time. However, the development of planning tools for these applications depends on the computer emulation of the underlying analyses presently performed by military planners, in particular, terrain analysis (Bonasso 1988, Powell 1987).

The operational planning process (CGSC 1986) includes mission analysis, terrain analysis, analysis of enemy dispositions and capabilities, and analysis of own troops. Terrain analysis focuses on the military aspects of the terrain and their effects on friendly and enemy capabilities to move, shoot and communicate. Military planners are taught to evaluate terrain in terms of observation and fields of fire, cover and concealment, obstacles, key terrain, and avenues of approach (OCOKA). To support computer based reasoning, data structures representing relevant features used by military planners need to be implemented. In terrain analysis, relevant features are extracted for use in the planning process, e.g. no-go terrain, mobility corridors, and avenues of approach. The computation of mobility attributes is straightforward. Mobility corridors are relatively open areas that avoid no-go regions and provide doctrinally adequate maneuver space. They permit relatively free movement and are classified according to the size of unit that

can maneuver along them. Avenues of approach are major axes into or through an area of operations and represent routes of movement for the attacker and places to be blocked for the defender. An avenue of approach for a given echelon is constructed from two or more mobility corridors for one echelon down that are close enough together, i.e. a brigade avenue of approach consists of two or more battalion mobility corridors no more than six kilometers apart.

The goal in terrain representation is to provide data structures corresponding to the conceptual objects reasoned on by terrain analysts and planners. While there are variations exhibited by individuals in reasoning on terrain structures, substantial agreement exists on the objects of relevance. This is perhaps due to the common teaching foundation for many Army officers. The reference used for the computer based terrain analysis described herein and the objects to produce for later use in planning is a student text at Command and General Staff College, "The Command Estimate". A major terrain feature of interest is the avenue of approach, which is constructed from mobility corridors, which are in turn derived by classifying the terrain into disjoint sets of no-go terrain and all other terrain. This hierarchy of data structures indicates an obvious approach to processing, although the algorithmic implementations are open to derivation. The approach taken was to use the definitions provided in "The Command Estimate" to generate the mobility attributes of terrain points. Then a planar partitioning algorithm was applied to calculate mobility corridors. The construction of avenues of approach from mobility corridors is still under development. Thus a basis of objects consistent with the problem domain is constructed to facilitate the emulation of the planning process, as it is described in terms relevant to terrain.

DATA PREPARATION

The available terrain data base is 100 meter resolution over a 97 by 125 kilometer region near the inter-German border (lower left reference point NA0063). Each terrain point has summary attributes as given in Table 1. Based on military doctrine, each point in the data base is classified as go, slow-go, or no-go terrain. Table 2 gives a set of criteria under which a terrain point is classified as no-go for armor and mechanized forces; if any criterion in the table is satisfied, the point is no-go. Algorithmic

TABLE 1. TERRAIN DATABASE ATTRIBUTES

Elevation	meters
Vegetation Height	meters
Urban	none, present
Hydrology	none, fordable river, non-fordable river, lake
Soil Type	muskeg, fine grained, coarse grained, heavy clay
Power Lines	none, present
Bridges	none, present
Land Use Code	open water, cropland, pasture, coniferous forest, deciduous forest, forest clearing, orchard or vineyard, dense brushland, open brushland, wetlands, peat cuttings, abandoned agriculture, bare ground or sand dunes, surface mines, urban
Road Type	none, autobahn, primary, secondary, trail
Obstacles	none, embankment or ditch, wall or fence, other manmade, military

TABLE 2. NO-GO MOBILITY CRITERIA

- Built up areas wider than 500 meters,
- Waterways that cannot be forded or spanned,
- Slopes of 45 percent or greater uphill,
- Trees more than six inches thick and less than 20 feet apart,
- Elevation variations more than 200 meters per kilometer,
- Minefields, tank ditches, barriers, or other obstacles,
- One trail per kilometer and no hard surface roads (except in open areas).

definitions corresponding to each criterion were devised for the mobility computations, with some criteria requiring estimation procedures, such as tree density and spacing (Bullock 1988). Figure 1 denotes the mobility attribute computations for a 52 by 52 kilometer subset of the terrain data (lower left reference point NA0488). This data is the basis for aggregation into no-go regions.

The aggregation process consists of collecting adjacent no-go points into regions. This process results in two classes of regions: "linear" features and "thick" features. Thick no-go regions have a minimum width greater than or equal to 300 meters, and linear no-go regions have width less than 300 meters. No-go regions are aggregated in a second pass; the output of the aggregation process is the set of no-go regions boundaries. A separation distance is chosen and regions with boundaries closer than the separation distance are combined under the rule that thick features can combine with thick or linear features, but linear features cannot combine with other linear features. This process eliminates non-nogo regions with width less than the separation distance, but may leave certain linear features unaffected. Next, boundaries of thick regions are smoothed by generating the convex hull of the region. Given a rationale that linear features can be readily breached and thick features are significant obstacles, the remaining linear features are ignored and the thick regions are assumed to represent the dominant areas of hindering terrain. Figure 2 shows the result of applying the aggregation process with a 300 meter separation distance to the data of Figure 1. The complement of these regions is the domain for mobility corridors. The boundaries of the thick regions are smoothed to remove local concavities and then input to compute mobility corridors. Planar decomposition methods are applied to this set of points to compute corridor lines and widths.

MOBILITY CORRIDOR GENERATION

Mobility corridor generation is based on two methods to partition the plane, Delaunay decomposition and Theissen triangulation (Renka 1981, Conway et al. 1988, Ohyaa et al. 1984, Shamos et al. 1975, Asano et al. 1985, Mehlhorn 1984, and Cline et al. 1985). With modifications, these partitions identify the essential elements for creating mobility corridors. These elements are a piece-wise linear curve identifying the "reference line" of the corridor and widths orthogonal to the reference line for each linear segment. The widths identify the corridor boundaries and enable the calculation of summary attributes.

Delaunay decomposition is a technique for dividing hyperspace into regions based on closest neighbors. It is defined as follows:

> Given a finite set of distinct points (nodes) in n-space, a Delaunay decomposition satisfies the following conditions

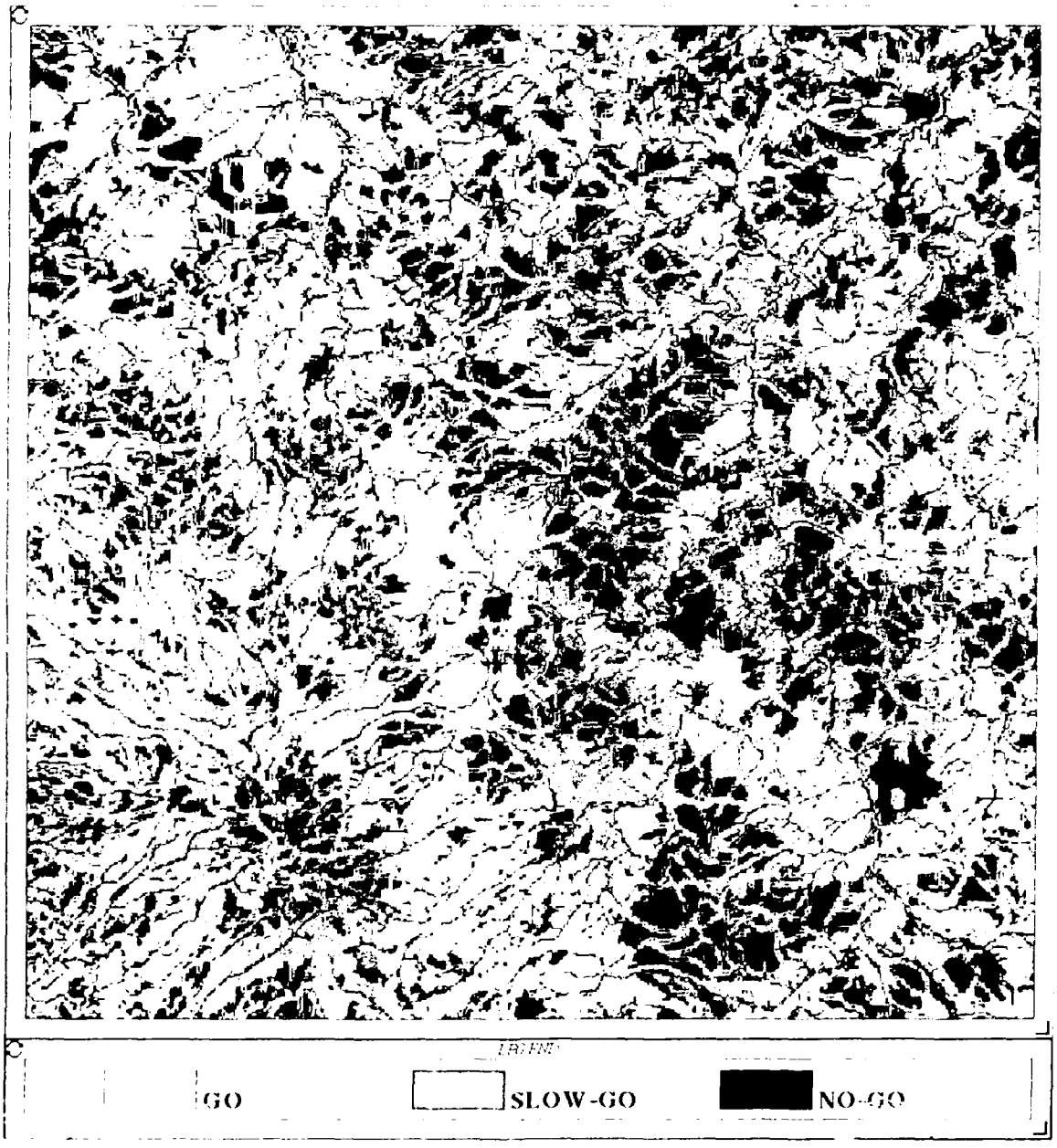


FIGURE 1. A 52 by 52 km terrain area with mobility attributes displayed.

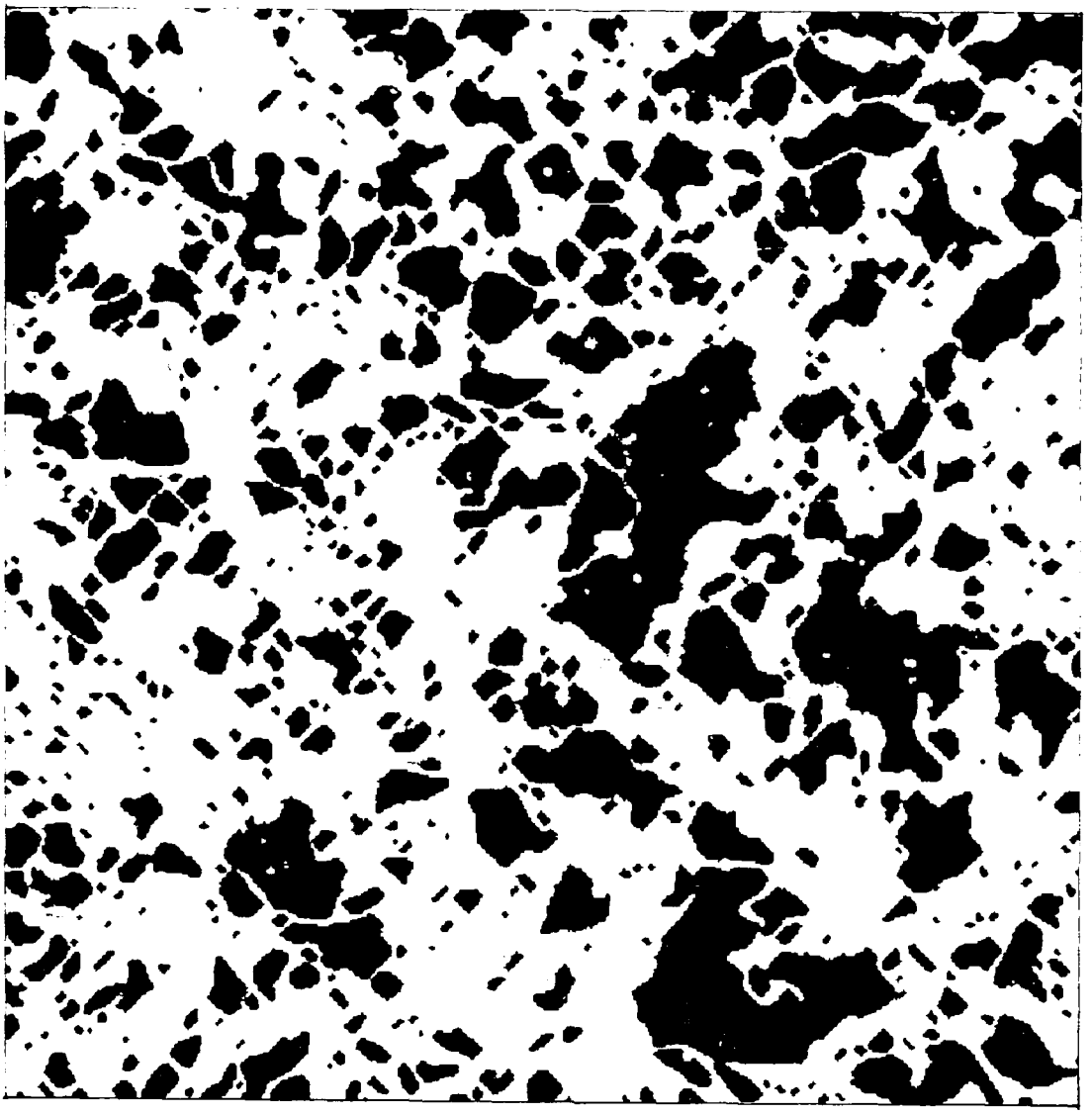


FIGURE 2. Aggregates of no-go terrain based on 300 m separation distance.

a. Each region R contains exactly one node;
b. The interiors of the regions are point-wise disjoint;
c. For any point P within any region R, the distance between P and the node within R is less than the distance between P and any other node.

For the purposes here, n-space is two dimensional; therefore the Delaunay decomposition simply divides the plane into regions of closest neighbors. Direct computation of Delaunay decompositions is non-trivial. However, solving a dual problem, Theissen triangulation, provides a more feasible computational approach. Theissen triangulation also partitions the plane and is defined as follows:

Given a finite set of distinct nodes in the plane, a Theissen triangulation is a partition of the portion of the plane contained in the convex hull of the set of nodes into regions, each one a triangle, such that:
a. Each region R, contains exactly three nodes, the vertices of the triangle;
b. The interiors of the regions are point-wise disjoint;
c. The union of the regions covers the convex hull;
d. For any region R, the circle which circumscribes the vertices of R contains no node in its interior.

It has been shown that for any Theissen triangulation, segments of the perpendicular bisectors of the legs of the triangles are the edges in the Delaunay decomposition for the same set of nodes (Renka 1981). Describing the computation of the starting and end points of these edges is facilitated by another definition: a triangle circum-center is the center of the circle that circumscribes the triangle. It is also the point where the bisectors of the legs intersect. Note that the circum-center need not be interior to the triangle. All three segments of the decomposition start at the triangle circum-center and extend "outward" until they intersect with Delaunay segments from other triangles. A more practical technique is to connect the circum-centers of all adjacent triangles out to the boundary of the area of consideration.

Now the partitioning algorithms can be applied toward generating mobility corridors. Consider the plane to be a military area of operations and the nodes to be the points defining the boundaries of no-go regions . Figure 3 depicts an area of operations

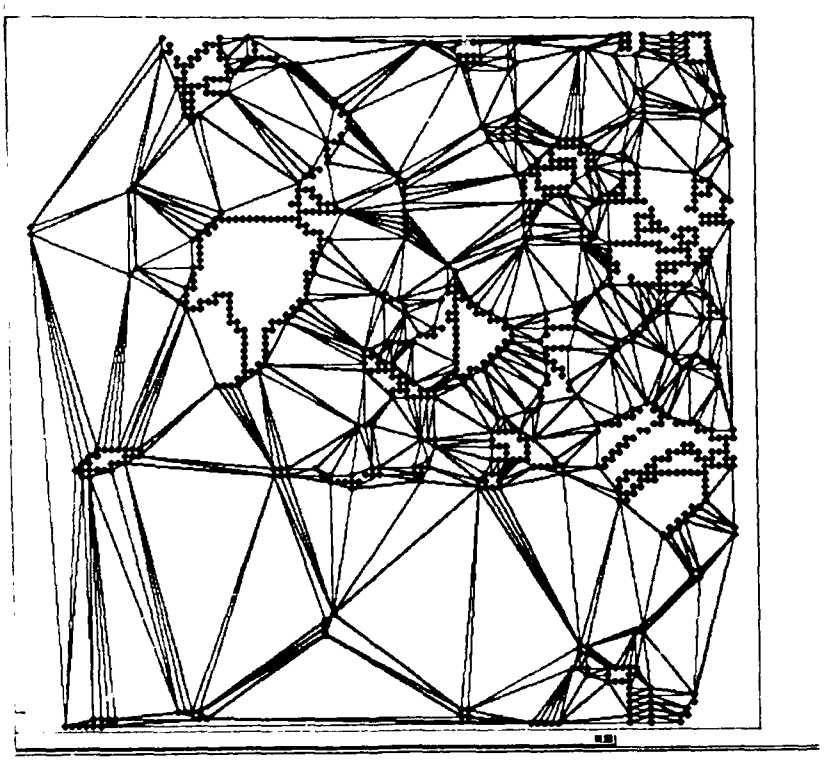


FIGURE 3. A 10 by 10 km terrain area with no-go aggregations and Theissen triangulation.

with the no-go regions calculated and aggregated for a 10 by 10 kilometer region near Lauterbach, Germany. The triangles of the Theissen triangulation within no-go regions are produced during the triangulation process, but are not very meaningful. These "no-go triangles" are eliminated from the solution set and the partition defined by the remaining triangles is also shown in Figure 3. The interior of the union of the remaining triangles define all mobility corridors in the area of operation. For each triangle, certain legs define the corridor width (Voronoi width), others define the boundaries of the corridors (Wright 1988). The piece-wise linear curve connecting the circum-centers, the Delaunay decomposition, is termed the reference line of the mobility corridor. The mobility corridor reference lines for the area of Figure 3 is shown in Figure 4.

This method produces all mobility corridors that are at least the width of the separation distance used in no-go region aggregation. Since the widths of all portions of the corridors are known, it is simple to select corridors based on the minimum width of a corridor to accommodate unit size, mission, or other criteria.

NETWORK GENERATION

It is necessary to construct a convenient data structure, in this case, a network, for capturing the information derived from the modified Delaunay decomposition to make it useful in a simulation. A network representation has several advantages which are important to the simulation. It is a form that is easy to use in both deriving routes, and simulating movement along a route. Secondly, it is an efficient means of storing the relevant information about mobility over the terrain area. The network must capture both the spatial structure of the mobility corridors, as given by the Delaunay decomposition, but also the mobility attributes maintained by the terrain points populating the corridors.

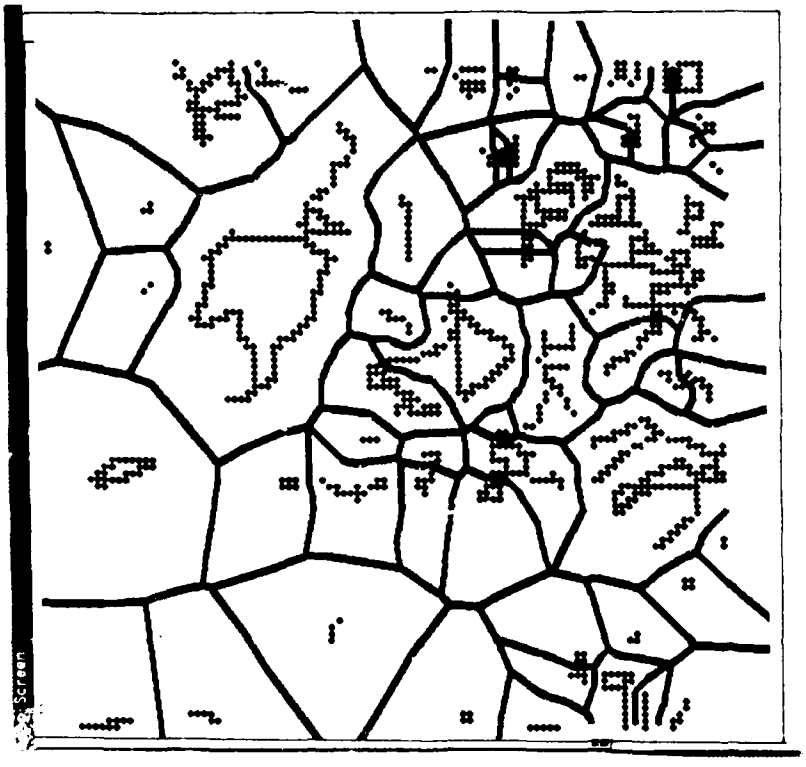


FIGURE 4. A 10 by 10 km terrain area showing the Delaunay decomposition.

A network is a collection of related nodes and edges. A node contains the pointers needed to represent the sequential ordering information, i.e. connectivity to adjacent nodes, while associated edges contain the attributes of the corridor. The creation of the mobility corridor network is accomplished starting from the information contained in the Theissen triangles. The network structure creation process has two stages. The first creates an intermediate structure capturing the detailed information of triangle circumcenter connectivity. The second stage uses the intermediate structure to generate a more compact, but less detailed representation of the corridor information.

During the Theissen triangulation, software objects are constructed to contain relevant information, given in Table 3. The first processing pass constructs intermediate objects, called tcenter objects, to define the mobility corridor domain, e.g. all no-go triangles are excluded. Each tcenter maintains data on adjacent tcenters, intervening distances, and associated Voronoi widths. This structure is a high resolution network of corridors with maximum fidelity to the underlying terrain. Since the simulation does not demand such detailed information, this structure is mapped to a more compact representation that contains summary attributes for the appropriate level of detail in the network.

In examining the intermediate network, it is obvious that a tcenter can only have one, two, or three adjacent tcenters, due to the nature of the triangulation process. Tcenters with one or three adjacent tcenters are termed junction nodes (hereafter referred to as nodes, see Table 4). Only tcenters with two adjacencies lie between nodes. The

TABLE 3. TRIANGLE OBJECT DATA

Name	symbol associated with triangle object
Node-List	counterclockwise list of triangle nodes
Arc-List	ordered list of arcs connecting the node-list
Voronoi-Widths	list of the widths of the Voronoi segments
Adjacent-Triangles	list of neighboring (continguous) triangles
Circle-Center	center of circumscribing circle
Circle-Radius	radius of circumscribing circle
No-Go-Triangle	boolean indicating membership in a no-go region
Tcenter-List	unordered list of qualified adjacent triangle centers

TABLE 4. NODE OBJECT DATA

Location	cartesian coordinates for node position
Adjacent-Nodes	list of nodes one edge away
Adjacent-Edges	list of edges that connect to each adjacent node

compact network representation eliminates these intervening objects and aggregates their information into summary attributes for an edge object that connects the nodes. The information contained in an edge is given in Table 5. In order to calculate attributes such as mobility or cover, references back to the original triangles are needed, as these attributes are estimated by sampling data within the triangle decomposition. The requisite back reference is the component-triangle-list, consisting of the sequence of triangle objects that define the mobility corridor. However, this list is only needed temporarily for the estimation process and is not included in the compact net representation. The output of the network generation process is a set of nodes and edges describing the mobility corridors in the terrain area.

NETWORK OPERATIONS

The primary capabilities a simulation unit will exercise on terrain are route selection and movement along the route. A unit may request a route to be generated based on user supplied policy. This policy defines a weighted evaluation of cost functions. Available cost functions include distance, traversal time, concealment, and cover, where traversal time is governed by movement rates specified for terrain with a given mobility attribute. The unit may also specify a minimum width that must be satisfied by all corridors in the route. Hence a company, doctrinally requiring 500 meters width for a mobility corridor, can request such a route. Similarly, a battalion can request a 1500 meter wide route. These requests are not guaranteed to be satisfied. If no route exists, or no route satisfies the unit's width specification, an error condition is returned. It is left to the requesting unit to decide an alternate policy, width, or destination if a route request cannot be satisfied. An A* search algorithm (Nilsson 1980) is used to determine the least cost route. Given a route, utilities can evaluate the route for each cost of interest. These functions enable the unit to access information relevant to selecting a route according to its needs and mission.

IMPLEMENTATION

The terrain processing features presented herein are implemented in C or Lisp. The mobility attribute computations and the terrain aggregation algorithm were written in C and were executed on a Sun 3/160 workstation. The mobility corridor generation program was written using the Flavors feature of Sun Common Lisp; it was executed on a Sun 4/260 workstation. Program execution time is reflected in Table 6.

TABLE 5. EDGE OBJECT DATA

Hi-Res-Line	sequence of circle centers connecting nodes
Hi-Res-Widths	sequence of Voronoi widths
Minimum-Width	minimum of hi-res-widths
Average-Width	average of hi-res-widths
Distance	actual distance along hi-res-line
Mobility	an estimate of the mobility for the edge
Cover	an estimate of available cover for the edge
Concealment	an estimate of available concealment for the edge

TABLE 6. EXECUTION TIME FOR MOBILITY CORRIDOR GENERATION

Area (Km^2)	Triangles	Nodes	Edges	CPU Hours
2704	31227	730	1000	5.75
100	766	26	26	0.04

CONCLUSION

Triangulation has been shown to be a viable method of finding mobility corridors using digitized terrain. The triangles form a natural data structure on which to calculate summary attributes of the mobility corridors. While corridors have application at least within combat simulations, they are only an initial step in the terrain analysis process. Future work should examine the calculation of avenues of approach and key terrain. Subsequently, the cognitive processes a simulation unit should use in determining routes and the value of terrain in light of its mission should be investigated.

ACKNOWLEDGMENTS

This work was conducted at Los Alamos National Laboratory in conjunction with US Army personnel under a memorandum of agreement with the Army TRADOC Analysis Command. Dr. Mark Johnson suggested the application of plane decomposition to the problem of mobility corridor generation and provided technical support in adapting the algorithms.

REFERENCES

Asano, T., Edahiro, M., Imai, H., Iri, M., and K. Murota, 1985: Practical Use of Bucketing Techniques in Computational Geometry. In G. T. Toussaint, (Ed.), Computational Geometry, Elsevier Science Publishing Company, New York, N. Y., pp. 153-195.

Bonasso, R. P., 1988: What AI Can Do for Battle Management, AI Magazine, 9. pp. 77-83.

Bullock, C. D., 1988: Methodology for Development of Digital Terrain Data for the Automated Model of Dismounted Combat, Technical Report GL-88-15, Geotechnical Laboratory, Waterways Experiment Station, Corps of Engineers, Department of the Army, Vicksburg, Mississippi 39181.

Cline, A. K. and R. J. Renka, 1985: A Generalized Delaunay Triangulation and the Solution of Closest Node Problems in the Presence of Barriers, unpublished paper available from Department of Computer Science, The University of Texas at Austin, Austin, Texas.

Conway J. H. and N. J. A. Slone, 1988: Sphere Packings, Lattices and Groups, Springer-Verlag, New York, N. Y.

Mehlhorn, K., 1984: Multi-dimentional Searching and Computational Geometry, Springer-Verlag New York, N. Y.

Nilsson, N. J., 1980: Principles of Artificial Intelligence, Morgan Kaufmann Publishers, Inc., Los Altos, Ca., pp. 74-76.

Ohyaa, T., Iri, M. and K. Murota, 1984: A Fast Voronoi-Diagram Algorithm with Quaternary Tree Bucketing, Information Processing Letters, 18, pp. 227-231.

Powell, Dennis R., 1987: Computer Based Terrain Analysis for Operational Planning, Proceedings of the 1987 IEEE International Conference on Systems, Man, and Cybernetics, pp. 1022-1026.

Renka, R. J., 1981: Triangulation and Bivariate Interpolation for Irregular Distributed Data Points, Ph.D. Dissertation, The University of Texas at Austin, Austin, Texas.

Shamos, M.I. and D. Hoey, 1975: Closest-Point Problems, Proceedings of the 16th Annual IEEE Symposium on Foundations of Computer Science, pp. 151-162.

Wright, J. C., 1988: Mobility Corridor Generation - A Different Approach, unpublished report, LA-CP-88-95, Los Alamos National Laboratory, Los Alamos, N. M. 87545.

U. S. Army Command and General Staff College, 1986: The Command Estimate, Student Text 100-9, Fort Leavenworth, Kansas 66027.

BETWEEN PROTOTYPE AND DEPLOYMENT: LESSONS LEARNED FIELD-TESTING AN EXPERT SYSTEM

Rosemary M. Dyer
Air Force Geophysics Laboratory
Hanscom Air Force Base
Massachusetts, 01731-5000

ABSTRACT

During the past two years, an expert system for forecasting the occurrence of fog has been installed at Seymour-Johnson AFB, North Carolina. Aside from validation of the rule base, this field test has provided valuable insights into some of the problems that must be addressed before such systems are deployed operationally worldwide. First is the necessity for a satisfactory trade-off between user involvement and real-time data input. In addition, varying degrees of sophistication on the part of the users must be taken into account. Systems which, like those used in meteorology, are geographically dependent, should be written in a form that makes them readily adaptable from one location to another. Adaptability includes the ability to use alternate data sources for those occasions or locations where standard observations and measurements are not available. This introduces the concept of maintainability, whereby an existing system can be updated to include new data sources and changes in the knowledge base. Each of these concepts is discussed in turn, with particular examples taken from the fog forecast system.

1. INTRODUCTION

In December 1985, the Air Force Geophysics Laboratory signed a one-year contract with GEOMET Technologies, Inc. for the development of a knowledge-based expert system designed to assist weather forecasters in predicting local visibility conditions. The object of the effort was to demonstrate the feasibility of using this approach, and the system that resulted was meant to be only a proof-of-concept prototype. Nevertheless, it was felt that evaluation of the success of the prototype required testing under near-operational conditions, with comments and suggestions from users who would be representative of weather forecasters within the Air Weather Service. Consequently, the initial prototype was deployed at three test locations (Dover AFB, Delaware and Seymour-Johnson AFB and Fort Bragg, North Carolina) shown in Figure 1. The results of these field tests, which concentrated on verifying that the knowledge bases produced accurate forecasts at each location, have been reported elsewhere (Stunder et al., 1987a and Stunder et al., 1987b).

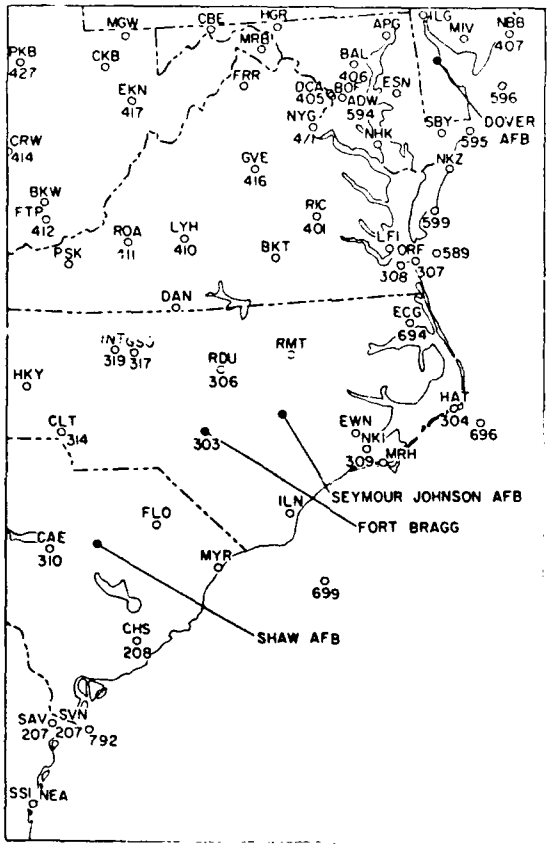


Figure 1. Locations for which fog forecast expert systems were developed. Zeus was developed for Dover, Seymour-Johnson, and Fort Bragg. The system designed for Seymour-Johnson was adapted to Shaw.

During the course of the contract, it became apparent that factors other than simple verification were equally important in evaluating the feasibility of deploying weather forecast expert systems at operational forecast offices throughout the Air Force. Additional evaluations were performed by Geophysics Laboratory personnel, in cooperation with Air Weather Service forecasters at Seymour-Johnson AFB. The lessons learned thus far during this project, which is still continuing, have ramifications for other expert systems destined for widespread deployment.

2. THE ROLE OF THE USER IN EXPERT SYSTEMS

Developers of expert systems, even more than standard computer programmers, must pay particular attention to the user-computer interface. The umbrella title, "user acceptance," includes numerous factors that contribute to the user's perception that the expert system provides a useful adjunct to available computer products and displays. Foremost of these, of course, is that the system produces advice that the user finds reasonable and acceptable. The field program demonstrated that there are many factors entering into this user acceptance/system accuracy equation. In hindsight, it might be argued that these results should have been

foreseen: in actuality, it is not possible to predict user reaction to an expert system without having the users evaluate the system under conditions closely resembling those in the field. Problems relating to usercomputer interaction fall into three categories: an incorrect conceptual model of the user; the method of user input; and inappropriate explanation facilities.

2.1 CONCEPTUAL MODEL OF THE USER

The system was designed to be used by apprentice weather forecasters, some of whom may be on their first assignment following forecaster training school. However, it was hoped that more experienced forecasters would also find the fog forecast expert system of some help. The developers of the system were research meteorologists; those interviewed were the more experienced operational forecasters at each location. Consequently, those who were the most likely beneficiaries of the system did not see it before being asked to exercise it in the field.

The first surprise to the researchers was the time it took to train the forecasters in the use of the system. The developers had apparently assumed a level of computer literacy only slightly lower than their own; they encountered more than one user who was completely unfamiliar with personal computers. The second surprise was the realization that queries asking for user input were not crystal clear, and the users asked for clarification. Finally, some of the user input required a level of sophistication in weather forecasting almost equal to that of making the fog forecast itself.

2.2 METHOD OF USER INPUT

In the initial prototype, all data were entered manually, and queries were phrased in a multiple-choice format. This caused immediate rebellion by the users. All the user comments the first week of testing included objections to the length of time it took to enter the observational data required to initialize the expert system, and the multiple-choice format of the non-data entries. This causes a bit of a dilemma. On the one hand, it is certainly not desirable to remove the user completely from the loop, for various reasons cited in the literature (Gordon, 1988). However, weather forecasting is a time-critical activity, and it is simply not feasible to have the forecaster chained to a system that constantly demands input. An appropriate compromise would be to have all numerical data (temperatures, pressures, wind speeds and directions, etc.) entered automatically, and other information, such as the present and future location of synoptic weather features, through a mouse or tablet. Further modifications, such as those suggested by Kemp et al., (1988) will also be tested.

2.3 EXPLANATION FACILITIES

All expert system shells advertise that they have an explanation facility. What this often consists of, as it did in this

instance, is a listing of the rules used by the system in reaching the given conclusion, or the rule for which the system is asking information. This often proved to be more exasperating than informative to users of the fog forecast system. Future plans call for a multi-layered explanation capability in the expert system, ranging from presentation of the rule being investigated, through a plain language statement of the line of reasoning being pursued, to a tutorial in the basic physics and meteorology underlying the system. Depending on the time available, and the user's inclination, these successive levels of explanation could be obtained by successive "Why?" queries. an explanation capability of this sort lies in the future, but it will probably be an essential part of the installation program we envision accompanying generic forecast expert systems to be deployed at Air Force weather stations around the world. In that case, once the system is installed, the installation program could serve as the tutorial program.

3. SYSTEM ADAPTABILITY

Even while the performances of the fog forecast prototypes were being evaluated, doubts were raised as to the feasibility of meteorological expert systems in a world-wide network. Obviously, some tailoring must be required to install the system at each location; however, if such tailoring requires extensive additional knowledge engineering, the whole concept of distributed expert systems becomes impractical. Over what geographic area is the prototype knowledge base valid, requiring only minor modification from site to site? How many such systems would have to be developed to cover all locations of potential interest to the Air Force? To evaluate this aspect of expert systems, we introduced the concept of adaptability, defined as the ease with which a system designed for a specific location can be modified for satisfactory performance at other locations.

3.1 MOVING FROM SEYMOUR-JOHNSON TO SHAW

As an initial test of its adaptability, the system designed for Seymour-Johnson Air Force Base was modified and evaluated as a predictor of fog at Shaw Air Force Base, South Carolina (see Figure 1). The modification was done without interviewing any of the forecasters stationed at Shaw, and the resulting system showed no significant difference in performance when compared with the Seymour-Johnson system. These results have been reported elsewhere (Dyer, 1988), and will not be discussed in detail here. The procedure used may be of some interest to those seeking to extend the geographic area over which their system is valid.

The first step was a thorough examination of the knowledge base. The version of the system at our disposal consisted of 207 IF-THEN rules, which can be grouped into the classifications shown in Table 1.

TABLE 1. ANALYSIS OF KNOWLEDGE BASE FOR FOG FORECAST SYSTEM

No Modification Needed:		
10 common sense	(5%)	
21 programming flags	(10%)	
32 basic meteorology	(15%)	(radiation fog)
Modification of Data Interface Sufficient:		
74 weather descriptors	(35%)	(synoptic map)
Modification Needed - Cast into Template Form:		
70 local rules of thumb	(35%)	(advection fog)

In all liklihood, most expert systems dealing with a geographically variable knowledge base will have rules falling into each of the categories listed here: the exact proportions will vary according to programming techniques and the subject of the expert system. In the present instance, only the 70 rules comprising the advective fog module of the system were expressed as local rules of thumb. These were transferred successfully to Shaw by first casting the rules of thumb into template form, then particularizing them to Shaw.

An example should serve to illustrate the template method. Fourteen rules dealt with surface wind directions favorable (or unfavorable) to the formation of advective fog. They were all of the form:

```
                IF      the month is BLANK,
                AND     the surface wind direction is
                        greater than XXX degrees,
                AND     the surface wind direction is
(RULE A)                less than YYY degrees,
                THEN    the surface wind direction is (is not)
                        favorable to the formation of
                        advective fog.
```

The critical wind directions were determined by consulting topographic maps, climatological data, and maps showing the location of meteorological stations. Rules that call for data from specific stations (for example, those upwind of Seymour-Johnson) were readily modified by changing the station designations to those more appropriate to Shaw (for example, those stations upwind of Shaw). There was no deterioration in the performance of expert system when it was used to forcast the occurrence of fog at Shaw AFB, rather than at Seymour-Johnson AFB. No attempt has been made to adapt the prototype as a regional system, but the experience of adapting it from Seymour-Johnson to Shaw indicates that there is nothing inherent in either the knowledge base or the architecture to prevent its adaptation to many locations along the eastern seaboard. However, it is also apparent that the template method of

designing regional expert systems cannot feasibly be expanded to a global system. It was decided that future weather forecast expert systems developed at the Air Force Geophysics Laboratory will be generic systems, capable of being installed by personnel at each site.

3.2 THE STRUCTURE OF GENERIC FORECAST SYSTEMS

The proposed generic systems will have the structure shown in Figure 2. Knowledge acquisition will delve more deeply into the knowledge base to determine what physical and meteorological principles underlie the rules of thumb. These will be common to all systems dealing with the same forecast problem, regardless of the location for which the forecast is to be made. For example, the advective fog module might consist of the following two rules:

	IF	a warm moist air mass is advecting towards the station,
	AND	this air mass will cool to its dewpoint
(RULE B)		when it reaches the station,
	AND	there are no countervening factors present,
	THEN	advective fog will form over the station.

and

	IF	advective fog has formed over the station,
(RULE C)	AND	there are no fog dissipating factors present or predicted
	THEN	low visibilities will persist.

Rules of this type would constitute the lowest level of the structure shown in Figure 2, and could be obtained directly from the scientific literature. This level would also include common-sense notions such as the fact that strong winds tend to dissipate fog, and that if it is night, the sun is not a factor in warming the air to a temperature above its dewpoint.

The second block in Figure 2 represents that portion of the knowledge base considering the effects of topography and of climatology. For example, this portion of a fog forecast system would contain the information that downslope winds are countervening factors in fog formation, and that during certain months, nearby large bodies of water are potential sources of warm moist air masses. This block would also contain regional climatology: statements to the effect that strong flow from the Atlantic Ocean is often a precursor of fog along the east coast of the United States and that inversion fogs occur along subtropical west coasts of Africa and North and South America.

These two lower blocks would comprise the expert system delivered to the individual forecast offices. The information contained in them can be obtained from textbooks and data bases. The next two blocks, containing local effects and individual rules of thumb, would have to be tailored for each location. Present plans call for the development of an installation program, with the

local forecaster entering in the latitude and longitude of the station, and the pertinent topographical information entered from a GIS data base. This could be supplemented by whatever climatological records are available for the site. Like the two lower blocks, this portion of the program (labeled "Local Climatology and Topography" in Figure 2) will remain relatively unchanged.

The top block of Figure 2 represents the local rules of thumb and standard procedures in effect at a particular site. The rules in this segment would be installed during interactive sessions with the local expert forecaster. Ideally, this can be done by having the expert answer computer queries similar to those that would be asked by a knowledge engineer. These questions would be framed in the general form "How do you determine whether or not the statement [an IF clause in a given rule] is correct?." Admittedly, this merely transfers the knowledge acquisition effort from a single team of knowledge engineers to the individual weather forecasters. This may not be a reasonable approach, in view of the lack of computer sophistication exhibited by the users testing the prototype. However, a properly designed installation program should allow the forecaster to tailor the expert system for a particular location. This would make feasible the deployment of versions of the system to multiple locations.

STRUCTURE OF A METEOROLOGICAL KNOWLEDGE BASE

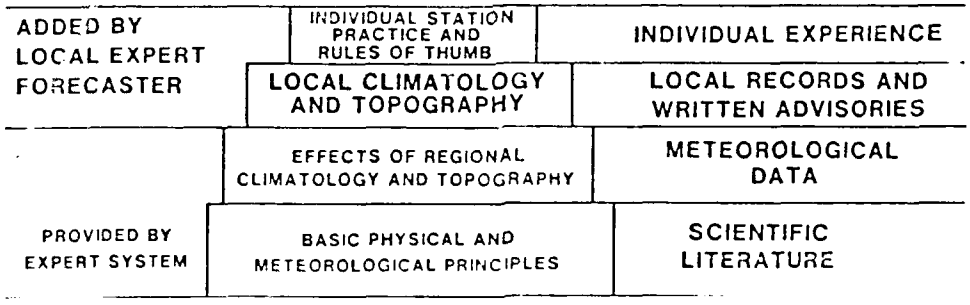


Figure 2. The structure of a knowledge base for generic expert systems in meteorology. The lowest layer contains general physical and meteorological principles taught in the classroom. Progressing upward, each level of knowledge becomes more particularized, culminating in the rules of thumb used by an individual expert at a given location. Typical sources of the knowledge contained in each level are shown at the right. The knowledge contained in the two lower blocks will constitute the generic system. The knowledge of the two upper blocks will be added by a local forecaster at each site.

An additional benefit of the architecture shown in Figure 2 is ease of maintainance. As new data collection and analysis facilities come on-line, they can be incorporated into the system through

modifications to the top block, without affecting the rules contained in the rest of the program. Similarly, as new discoveries are made in meteorology, the rules of thumb contained in the upper block will change, also without affecting the remainder of the system. Thus, adaptability will be achieved over time, as well as location.

4. CONCLUSIONS

Field tests of the fog forecast expert system has lead to two main conclusions. First, much more attention must be paid to the user interface. Graphics, mousing capability, menus, light pens, and explanation facilities should all be utilized to make future expert systems as user-friendly as possible. This is not, properly speaking, in the realm of artificial intelligence. Nevertheless, the success of any expert system under battlefield conditions may well depend on how quickly someone unfamilar with the system and with only minimum computer literacy can be taught to use it.

The second lesson learned from these tests is that expert systems written for geographically varying domains such as weather forecasting must be adaptable to multiple locations. Expert system use has progressed beyond the proof-of-concept stage, where an expert system that does a task well is considered a success. Now we are entering a second stage, where hard looks are taken at the time and effort invested in knowledge acquisition and computer program development. If the resultant systems are applicable to only a single location, or a single narrow task, they may not be worth the development effort.

REFERENCES

Dyer, R.M., 1988. The Adaptability of Expert Systems in Meteorology: Fourth International Conference on Interactive Information and Processing Systems in Meteorology, Oceanography and Hydrology, Anaheim, California, February 1-5 1988, pp. 257-8. American Meteorological Society.

Gordon, S. E., R. T. Gill, and T. A. Dingus, 1987. Designing for the User: the Role of Human Factors in Expert System Development. AI Applications 1 #1 pp. 35-49.

Kemp, R., T. Stewart, and A. Boorman, 1988. Improving the Expert System Interface: AI Applications 2 #4, pp. 48-53.

Stunder, M., R. M. Dyer, and R. Koch 1987a. The Use of an Expert System in Assisting Forecasters in Visibility Predictions: Third International Conference on Interactive and Processing Systems in Meteorology, Oceanography and Hydrology, New Orleans, Louisiana, January 1987, pp.206-7. American Meteorological Society.

Stunder, M. J., R. C. Koch, T. N. Sletten, and S. M. Lee 1987. ZEUS: A Knowledge-based Expert System that Assists in Predicting Visibility at Airbases. AFGL-TR-87-0019. Air Force Geophysics Laboratory, Hanscom AFB, MA 01731-5000.

COMPUTER DETECTION AND TRACKING OF MULTIPLE OBJECTS IN TELEVISION IMAGES [1]

Andrew Bernat, Stephen Riter, and Darrell Schroder
The University of Texas at El Paso
El Paso, Texas 79968, USA

ABSTRACT

The automatic detection and subsequent tracking of moving objects is an important area of application of computers and artificial intelligence in the battlefield. This paper discusses our efforts to develop a system which will automatically detect and determine trajectories for multiple objects in a sequence of television images. Particular attention is paid to the development of techniques suitable for objects exhibiting low-contrast against a complicated, time-dependent background. The front-end of the system consists of 16 parallel microprocessors which each operate on a segment of the total video image to determine if there has been change in that segment of the image. Change is detected based upon a novel scheme for calculating the median pixel value in the segment and comparing it with the median value from an earlier frame. Motion detection and subsequent object tracking are performed by assuming that objects exhibit smooth motion on the time scale of image acquisition. There are no other requirements as to the form of object motion. This paper discusses the algorithms used and our practical realization of these algorithms in a complete system.

1. BACKGROUND

Imaging devices such as television or infrared cameras are used in a variety of applications to detect and track object motion. On the battlefield such techniques could be used to acquire and track a wide range of objects from soldiers to airplanes. Another possible military application of such techniques is in security systems where the cameras are used to monitor access to controlled facilities such as nuclear plants or airfields.

The application which stimulated this study grew out of an effort by the U.S. Immigration and Naturalization Service (INS) to increase the effectiveness of imaging systems as an aid to monitoring traffic across international borders. INS is responsible for detecting and preventing the illegal entry and smuggling of aliens into the United States. INS has established a test bed in the El Paso area to test

[1]This work was supported by the National Institute of Justice, U.S. Department of Justice, through grant number 84 IJ CS 0041.

the use of closed circuit television and infrared cameras as a means of improving the effectiveness of Border Patrol agents involved in linewatch activities. The initial configuration of the system consists of eleven low light level TV cameras placed at nine locations along a nine mile segment of the border. It is estimated that there are nearly 20,000 illegal border crossings per month in the El Paso area. For a description of the INS project see Bernat et al. (1987).

2. IMAGE CHARACTERISTICS

The scenes viewed by these cameras are extremely complicated. There are foreground objects from the urban environment of El Paso, background objects from the urban environment of Juarez, and a tremendous number of moving objects in both cities. These objects are of no inherent interest but do serve to provide a complicated, changing and confusing background upon which the objects of interest move. The scenes are characterized by fuzzy edges, a lack of contrast, noise and wind induced camera motion. Furthermore, because of the outside location, there is an almost continual variation in illumination due to clouds and other environmental factors. Thus the types of images motivating the INS study are also representative of battlefield conditions (although considerably less threatening). Ideally a motion detection system must be able to discriminate between all of the sources of possible change and the events of interest.

3.0 CHANGE DETECTION

3.1 FRAME DIFFERENCING

The initial approach to motion detection adopted (in hardware) was to use a system which combined frame differencing with averaging over selected geopixels (a rectangle of contiguous pixels). The system captured one image per second, digitized the image and stored it in an array. The image was then differenced with the next image captured. The differences were than averaged over selected geopixels. If the average exceeded a predetermined threshold, an alarm was sounded. The selection of the threshold and the area was manually set by the operator.

Due to the complicated nature of the images as described above, it should not be surprising that this simple approach of geopixel averaging turned out to be unsatisfactory. Operators would set the threshold and then be plagued by unacceptably high false alarm rates. The operator would then raise the threshold until the false alarm rate would be lower, however the detection rates would then become unacceptable.

3.2 MEDIAN COMPARISON

To obtain immunity to noise a variety of other techniques have been proposed

including averaging or weighted averaging over areas of a frame (Sugimoto et al., 1968) and algorithms based upon maximum likelihood detection schemes (Yakimovsky 1976). The former are simple but still susceptible to noise while the latter, although relatively robust, are computationally burdensome, difficult to analyze and not realizable in real time. We have studied computing a median or mean over limited areas of a frame (Bernat et al., 1988). This quantity is then differenced with the mean or median from the preceding frame. If the result is greater than a threshold, then motion has occurred.

The median is particularly attractive and follows logically from the use of median filters in edge detection. It is known, for example, that median based edge detectors are relatively immune to spike noise yet respond quickly to changes in intensity (Gallagher and Wise, 1981). If motion detection is visualized as a type of edge detection between frames, then it follows that median based motion detectors should exhibit similar properties. Using synthetic images we have been able to demonstrate that median based motion detection is an effective technique when processing must be done in real time (Bernat et al., 1988).

Another attraction of a median based motion detector is its computational efficiency. The median can be determined by sorting all of the values in a particular area and then selecting the middle value. Traditional sorting routines require a considerable number of computational cycles. The median can be found more quickly by utilizing the fact that, in a digitized image, each grey level value is one of a finite set of numbers, typically 128, 256 or 512, and allocating an equivalent number of memory cells to the process of determining the median. The cells are all initially set equal to zero and assigned an address between 1 and the number of grey scales. The computer then examines each pixel in the area and increments the cell associated with the grey scale encountered at each pixel by one. When this process is completed the computer begins with the first cell and sequentially adds the values encountered. When the sum reaches one half the number of pixels the address of the cell when this is encountered is the median.

3.3 HARDWARE IMPLEMENTATION

We have realized the motion detection system with the configuration illustrated in Fig. 1. It consists of a video digitizer, an array of parallel microprocessors and a timing and control system. After each frame is digitized, each microprocessor asynchronously determines the median grey level for its portion of the image. This grey level is compared to previously determined ones and a decision made about the likelihood of change. The entire operation is observed by a central processing unit which outputs information as detected and extracts information from which decisions can be made. The output from the central processing unit is a binary image representing those areas of the original image undergoing change.

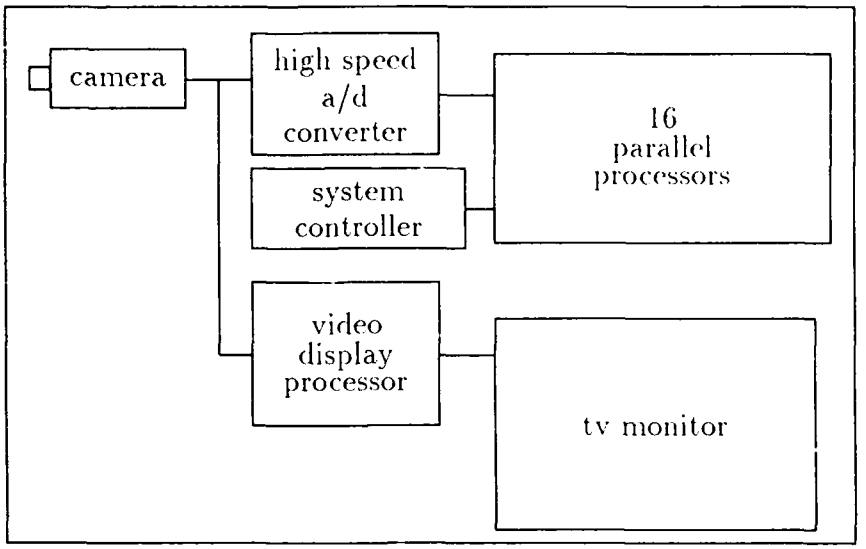


Figure 1. System configuration.

When designing the system, our goal was to develop an approach which combined high speed through parallel processing techniques with absolute minimum cost. This required implementation utilizing existing 8-bit hardware. Thus we were able to replace a single fast but expensive processor with many processors which individually are much slower, but in aggregate represent a much lower total system cost and a net higher processing speed. Current estimates suggest that a signal processing element can be constructed for less than $50 and an array of 16 such processors with appropriate algorithms will process the video information at least four times faster than a system with a single 80286 processor. A complete commercial system excluding the video camera should cost less than $1000 and be capable of detecting the portions of the video image that are changing in near real time.

4.0 MOTION DETECTION

Detecting changes in an image represents the first stage of our system. As discussed above, change may be caused by many factors including both environmental factors and object motion. The second stage of our system separates out true object motion. We present here two techniques for detection and tracking of object motion. Both are broadly based on the fact that, as an object moves across the screen, motion in one area is related to motion in adjacent areas.

4.1 ADJACENT CELL APPROACH

If each area of the screen is able to communicate its decision concerning motion to its neighbors, then it should be possible to obtain an enhanced estimate of object motion. We have presented simulation studies of a system in which a one

dimensional array of elements communicating with their neighbors was tested for immunity to noise in Riter et al. (1988). It is apparent that the array is capable of accurate motion detection even in the presence of a high noise level. This adjacent cell approach worked quite well and additionally generated the trajectory of the object motion if the time history of the array outputs is retained.

The difficulty with this approach comes when we attempt extension to two dimensions. The number of neighbors with which communication is necessary is now at least eight. Of course many of these connections can not physically occur, e.g., motion up a tree or building. The difficulty becomes: How do we teach our computer vision system to recognize the valid object trajectories and to ignore the rest?

4.2 COHERENCE APPROACH

Because of the computational difficulties described above, we have also explored the possiblity of motion detection and tracking using the concept of path coherence (Sethi and Jain, 1987). Path coherence is a measure of the "smoothness" of a trajectory. According to Sethi and Jain (1987), this smoothness is a function of velocity, i.e., an object's acceleration should remain constant if images are obtained sufficiently frequently. To this requirement of constant acceleration, we add coherence in grey scale, i.e., an object's radometric properties should also remain approximately constant between images. However, the algorithm presented in Sethi and Jain (1987) is not suitable for our application because it requires noise free images, all trajectories must be present from the first through last images, and it requires multiple passes through the trajectory points as each new image is acquired. This procedure is not acceptable for real time applications.

Recall that the input to the tracking portion of the system consists of a binary image representing the changing pixels. As a new image is input to the trajectory determination phase, the notion of path coherence is used to determine which changed pixel best represents the continuation of each trajectory. In cases with false change pixels due to image noise, more than one point may satisfactorily continue a trajectory; all of these continuations are retained. We then use a pruning technique to eliminate unsatisfactory trajectories. All of the change pixels in the latest image which are not included on a trajectory are considered as possible starting points for newly arising trajectories. Trajectories which cannot be satisfactorily continued are considered terminated. By comparing these terminated trajectories and the newly created ones, we handle the case in which a truly changing pixel is lost in the noise. The output of the system is a list of possible trajectories which satisfy the concept of path coherence. In order to control the number of possible trajectories, we also prune this set of projected trajectories.

The algorithm requires specification of the coherence weights corresponding

to speed, direction and grey scale, and the severity of pruning. These weights should be chosen to fit the expected trajectory characteristics. There is a trade-off here--the looser we make our requirements for coherence or the less trajectories we prune, the less likely we miss a truly moving object, but the greater the computational burden. For the simulated trajectories shown in Fig. 2, we have the results shown in Figs. 3 through 6.

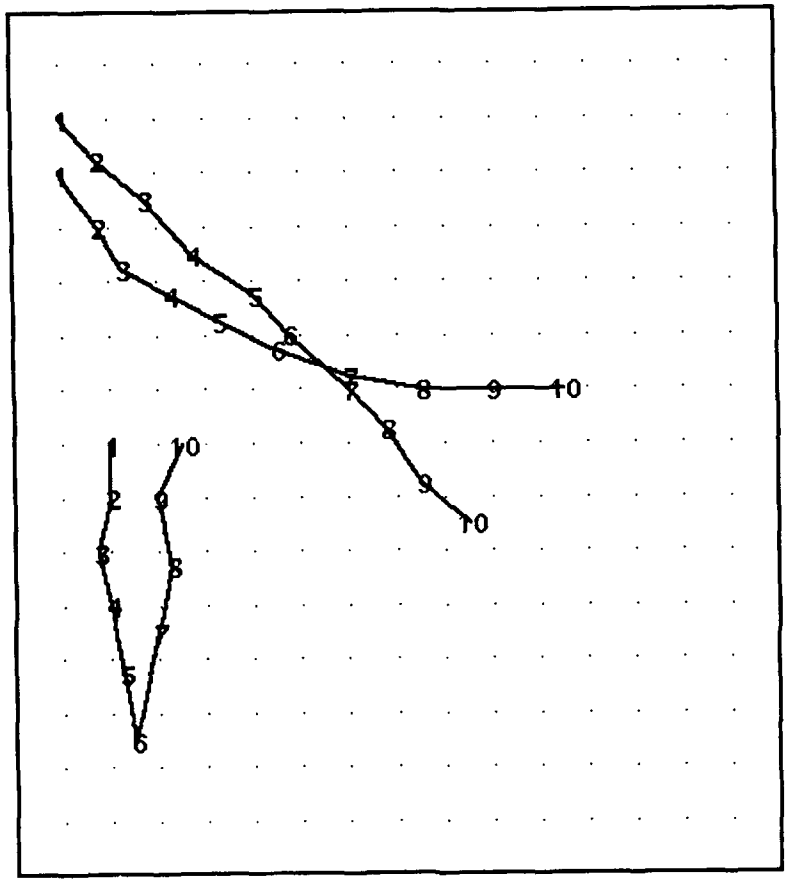


Figure 2. Three simulated trajectories. A sequence of ten images has been compressed into this figure; the numbers define the image in which each point exists.

5.0 CONCLUSIONS

We have demonstrated the feasibility of detecting and tracking moving objects in a sequence of television images in a laboratory setting. The first stage, change detection, is presently being implemented in hardware. The second stage, motion detection and object tracking, has been studied through simulation studies. Further experience with actual image sequences is required in order to provide guidance for the choice of the pruning parameters.

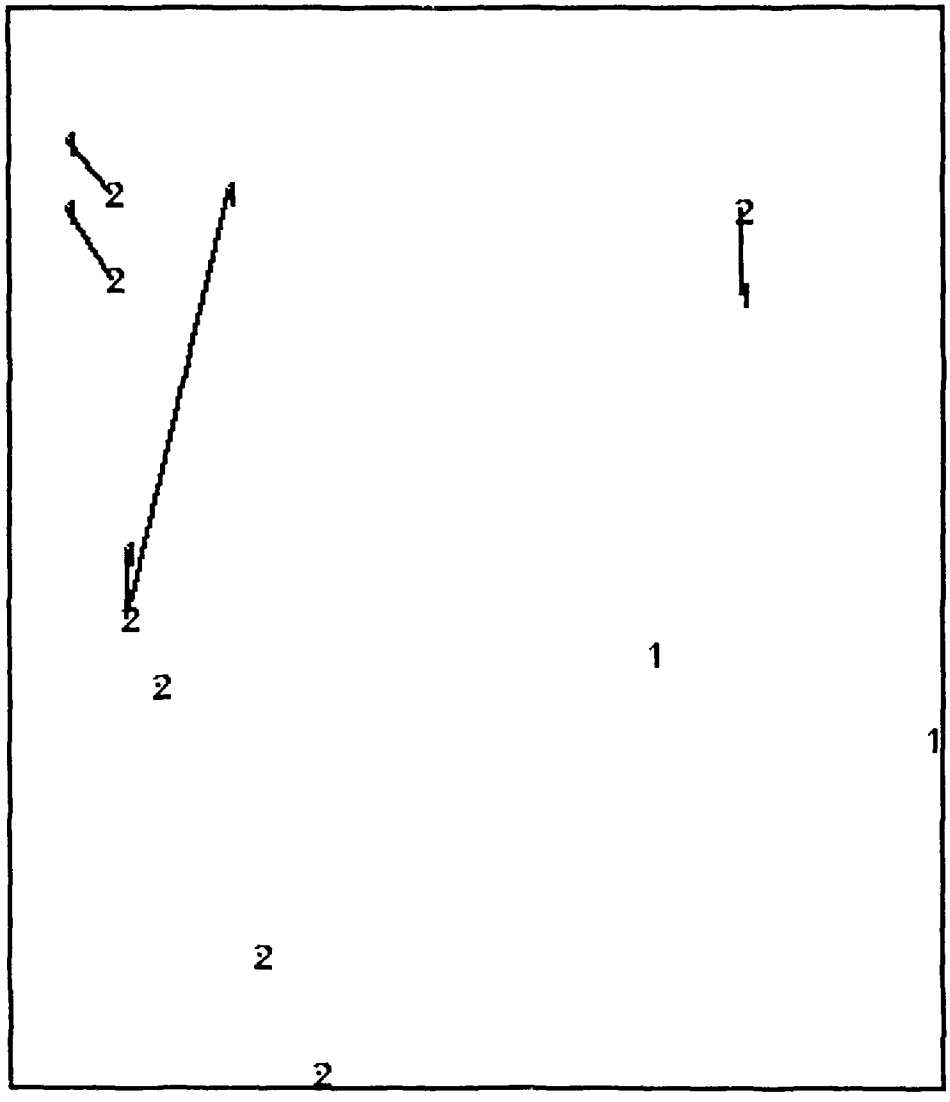


Figure 3. The detected trajectories after image number 2 has been processed. The additional points in this figure (relative to Fig. 2) represent noise hits which were added with a probability of 1% per geopixel. In order to initiate the trajectories, all possible combinations of points in images 1 and 2 are used, subject to simple boundary conditions.

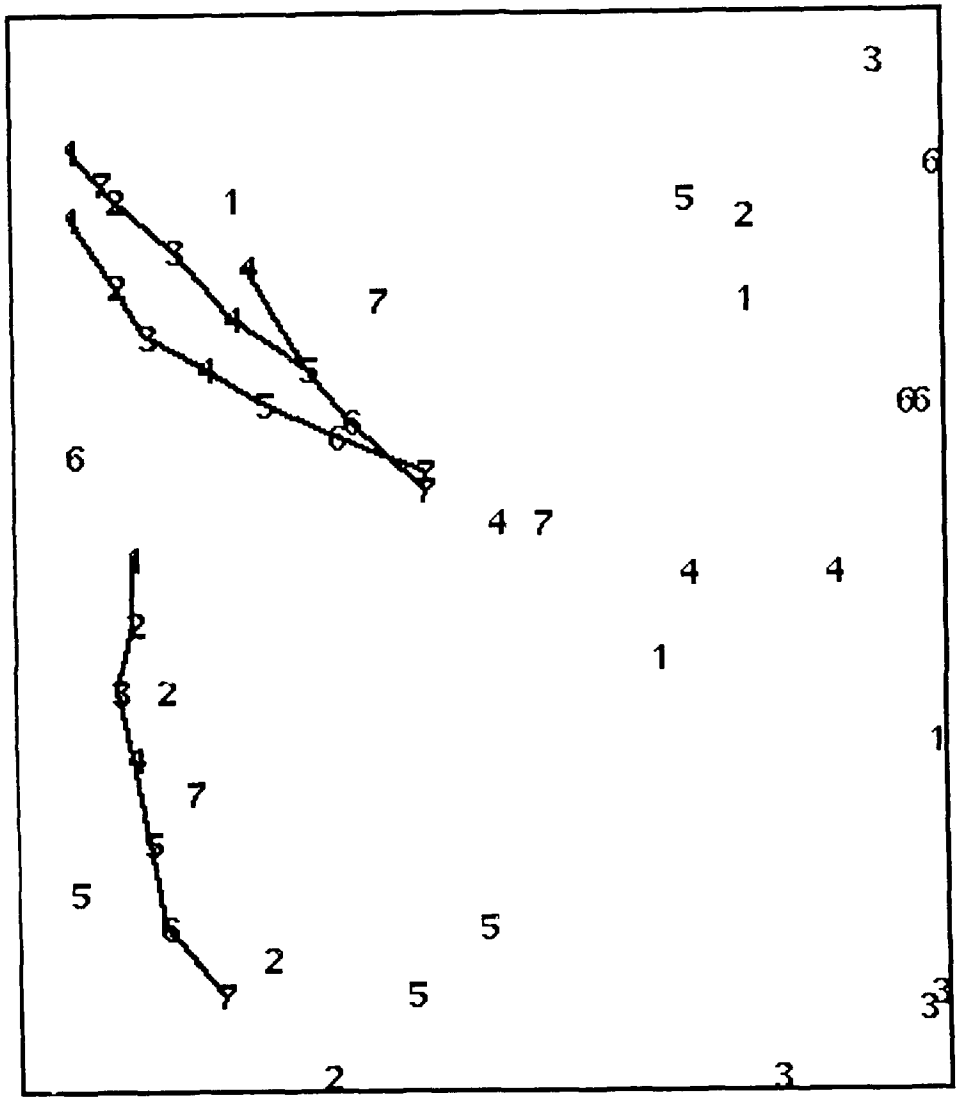


Figure 4. The detected trajectories after image number 7 has been processed. Note the extra (false) trajectory. This trajectory satisfies the path coherence condition: a different choice of relative coherence weights would eliminate it.

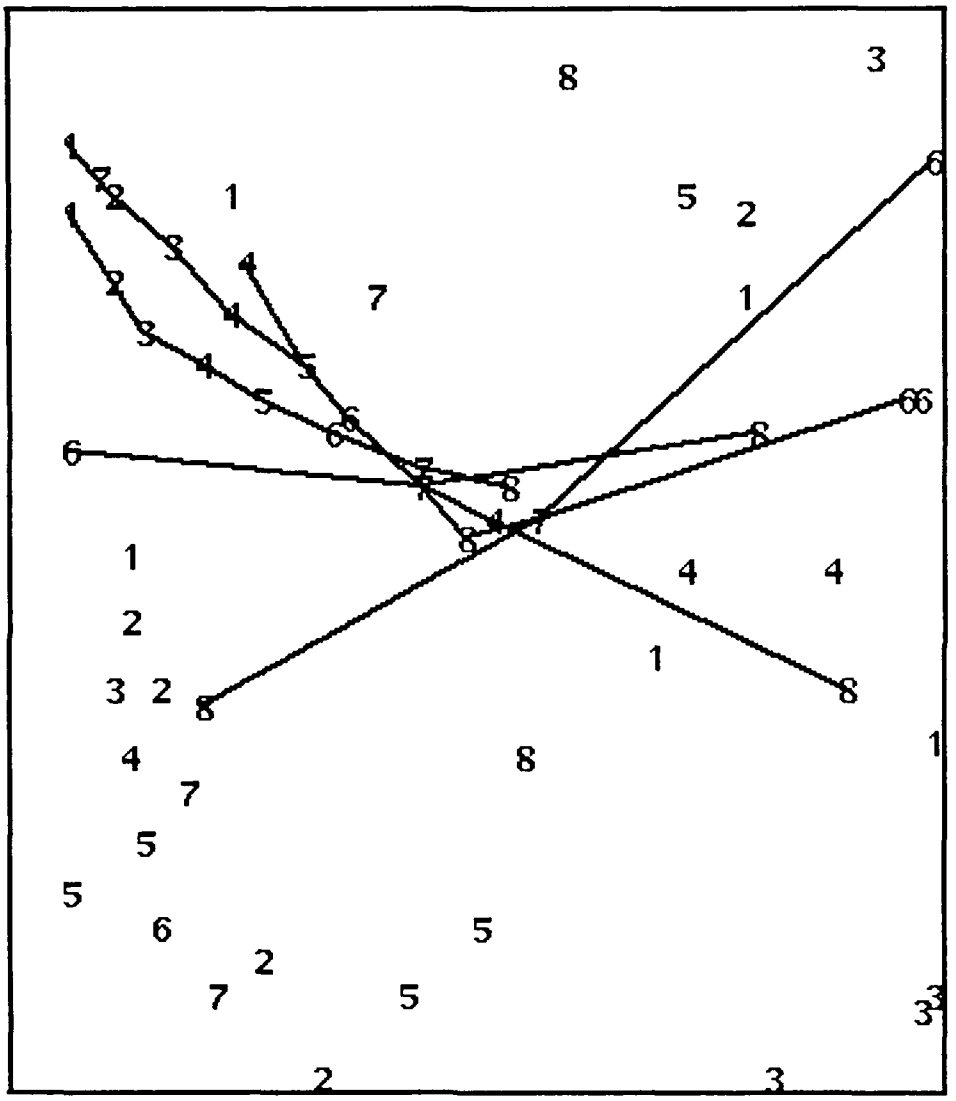


Figure 5. The detected trajectories after image number 8. At this point one true trajectory has been lost due to the large change of direction. There are now a considerable number of false trajectories.

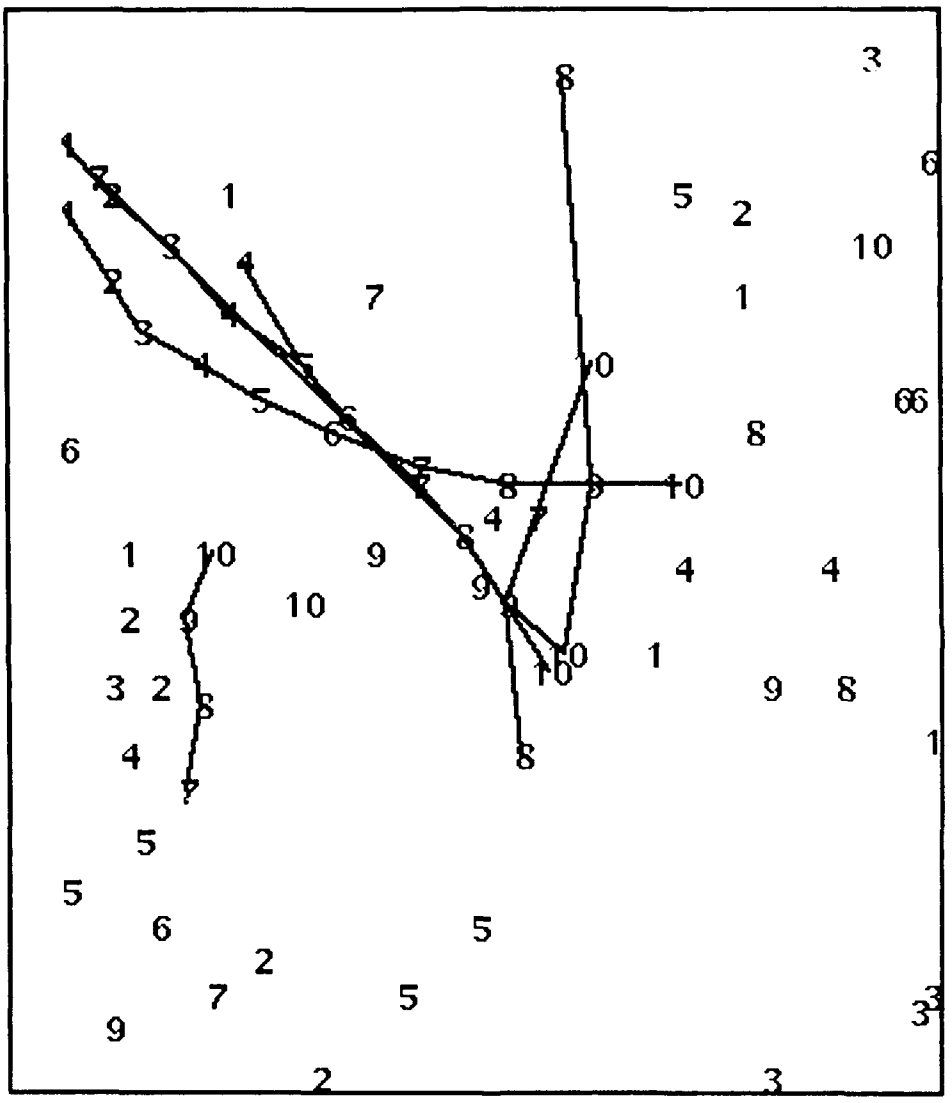


Figure 6. The detected trajectories after image number 10. Note that the trajectory lost after image 7 has been recovered. With the exception of the trajectory described in Fig. 4, all of the false trajectories have a length of 3, which could be a basis for pruning.

REFERENCES

Bernat, A. P., J. Nelan, S. Riter, and H. Frankel, 1987: Security Applications of Computer Motion Detection. In J. F. Gilmore (Ed.), *Applications of Artificial Intelligence V, Proc. SPIE*, **786**, 512-517.

Bernat, A. P., S. Riter, and J. Nelan, 1988: A Comparison of Three Motion Detection Techniques, *Optical Engineering*, **27**, 524-527.

Gallagher, N., and G. Wise, 1981: A Theoretical Analysis of the Properties of Median Filters, *IEEE Trans. on Acoustics, Speech and Signal Processing*, **ASSP-29**, 466-471.

Riter, S., A. P. Bernat, and D. Schroder, 1988: Computer Detection and Tracking of Moving People in Television Images, *Proc. of the IEEE 1988 Conference on Systems, Man and Cybernetics*, 1013-1016.

Sethi, I. K., and R. Jain, 1987: Finding Trajectories of Feature Points in a Monocular Image Sequence, *IEEE Trans. on PAMI*, **9**, 56-73.

Sugimoto, S., H. Matsukis, and Y. Ichioka, 1968: Implementation of Tracking and Extraction of Moving Objects in Successive Frames, *Applied Optics*, **25**, pp. 950-996.

Yakimovksy, Y., 1976: Boundary and Object Detection in Real World Images, *J. ACM*, **23**, pp. 599-618.

SPATIAL AVERAGING OF SOIL MOISTURE*

Perry J. LaPotin
Department of Physics & Astronomy
and
Department of Earth Sciences
Dartmouth College
Hanover, New Hampshire 03755

and

Harlan L. McKim
Research Division
U.S. Army Cold Regions Research and Engineering Laboratory
Hanover, New Hampshire 03755-1290

ABSTRACT

Soil moisture will travel along the path of least resistance in an effort to conserve energy and momentum. The process dynamics are a function of the soil characteristics, hydraulic regime, and meteorological factors that change in time and space. In this paper, a new approach for the development of a spatial averaging algorithm is presented. The method is outlined using a new symbolic language and an equivalent computer code. The enclosed algorithm determines available water for routing, and spatially averages resultant moisture levels to determine flow direction and flow velocity. The method uses real-time data to supplement the often sparse information from in-situ sensors and integrates the information into the hydrodynamic routing algorithm. Results indicate that spatial averaging is a powerful and computationally efficient method to simulate hydrologic routing over variable terrains with dynamic meteorological conditions.

1. INTRODUCTION

Early efforts to average spatially dependent samples used test hole or "core" information to determine a "Kriged" estimate of the localized variation (Matheron [1963,

*Funded under contract DACA-89-87-K-008 Department of Physics and Astronomy, Dartmouth College, Hanover, N.H. 03755.

1965, 1970], Krige [1966]). Matheron applied the Kriging approach to the local estimation of soil type, and along with other authors, produced soils maps with varying degrees of resolution (Burgess et. al. [1981], McBratney and Webster [1981], Webster and Burgess [1980], and Clark [1982]). In each case, static semivariogram (Kriging) or semicorrelogram (Scale) methods were used to determine estimates of regional variation. These methods proved satisfactory for regions with slowly evolving hydrology and near steady-state meteorologic conditions (e.g ore and stress-fault analysis), but failed in their estimation of soil moisture under evolving (non-steady state) climate conditions (Yeh and Yoon [1981], McBratney and Webster, [1983]). The failure was due in part to: (i) the dynamic nature of the state variable- soil moisture- whose behavior changes in time and space, and (ii) the static characterization of soil moisture from a priori estimates of regional variation (using the semivariogram or semicorelogram as the basis for the estimate). McBratney and Webster, [1983] summarized the problems associated with the usage of semivariograms to model dynamic, non-stationary, phenomena:

> "It is not wise to assume that the property of interest is stationary over the whole region or that a given form of local trend applies everywhere...The semivariogram is usually known accurately only over small neighborhoods within a region and will almost certainly not be known for lags approaching the distance across the region"

To reduce the problems associated with regionalization of Kriged samples, Journel and Huijbregts [1978] illustrated that "a global estimate obtained by Kriging over the whole region" is equal to "the average of local estimates made for small neighborhoods." Hence, point estimates could be weighted by their respective areas and summed to produce a regional estimate of the average state condition. However, the method was shown to be insufficient for estimating the regional variance since estimates at neighboring points are seldom statistically independent (Chirlin and Wood [1982], Russo and Jury [1987a,b]).

Scale estimation methods superceded Kriging techniques by incorporating the robustness of a general linear model, and the correlational dependence between sample points to study "the average distance over which the spatial variations in the (state variable's) properties are correlated" (Papoulis [1965], Russo and Jury [1987a,b], Gelhar and Axeness, [1983]). By applying correlational measures, the estimates are scaled (-1,1) and are thus independent of the units of the original measure. If assumptions are made concerning the distributional properties of the random variable, e.g. $\theta(x)$, a random variable in x with an approximate two parameter Normal distribution ~ $N(\mu,\sigma^2)$, then a wide family of estimators (with varying levels of precision) may be derived for both the population mean and the population variance.

Kitanidis [1983] summarized the uses for Scale estimation in the modeling and simulation of soil moisture :

"Scale plays an essential role in problems which require estimating spatial variations of a given soil property from limited observation points and in the solutions of transport problems in heterogeneous media using stochastic modeling."

Problems with the approach correspond to those previously outlined for the standard Kriging method: (i) uncertainty resulting from less than optimal selection of the number and location of sampling points, and (ii) a correlogram construction based solely upon the uncertain sample size and sampling locations invariant with time, and (iii) an abundance of assumptions required to obtain "practical" solutions for the regional variance term.

In recent studies, State estimation methods have been applied to ameliorate the difficulties associated with Kriging and Scale methods (Schweppe [1973], Gelb [1974], Olea [1974], Anderson and Moore [1979]). These methods allow for the simple inclusion of autodynamic variables within robust methods for the solution of the corresponding systems of simultaneous equations. Chirlin and Wood [1982] outlined the close association between Universal Kriging techniques and simple state-space theory:

"Universal Kriging, and indeed any method to identify the form of the sampled, admittedly nonstationary, random process, must begin by restricting its attention to a workable subset of the set of all nonstationary random functions...The reworking of the Universal Kriging equations into state space form allows for the straightforward sophistication of the process and observed models, should that be desired."

In each application of state theory, the approach has been to refine the basic Kriging and Scale estimation techniques to allow for the specification of various attributes in the sampled region[1]. In the following brief, a more generalized application of state-space theory is proposed that uses symbolic notation to derive the system of differential equations required to model complex phenomena. The approach uses four basic operators to define the system, and relies on "object-oriented" programming to allow for easy access and customization of the system to real-time data. In the examples provided, the theory is applied to the modeling of volumetric soil moisture in multiple soil horizons.

2. SYMBOLIC APPROACH

In the development of a symbolic representation for state-space theory, the early efforts of Forrester [1968] and Richmond[1985] have been expanded to handle multi-dimensional modeling under conditions of real-time data (interlaced real-time data). Early applications of the Forrester method applied state modeling, under simple univari-

[1] For example, the addition of observational error (Schweppe [1973]), and the inclusion of dynamic joint evolution states (Gelb [1974], Chirlin and Wood [1982]).

ate conditions, to model complex systems (i.e. interation in one variable). With the advent of matrix notation and vector operators, multidimentional iteration was possible, and state-space models could be examined. In the example that follows, a simple mass balance (continuity) model is derived using the notation of: (1) standard control volume theory, (2) corresponding differential equations, and (3) state-space symbolic notation. *The continuity model illustrates how stepwise differential models may be constructed, in* symbolic form, without relying on a "closed-form" representation. In addition, the equivalent systems representation (Figure (2)) introduces the reader to the symbolic notation used to derive the hydrodynamic model and spatial averaging *shell* in later sections. In Figure (3), a "skeleton" system for simple routing of volumetric soil moisture is provided. Although abbreviated for clarity, the figure illustrates how a basic hydrodynamic routing algorithm is constructed and organized according to rate processes.

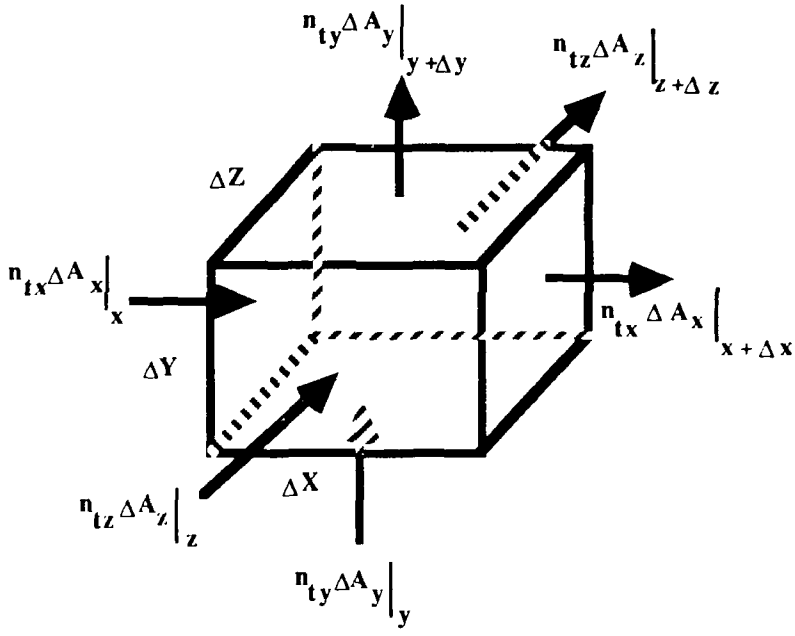


Figure 1. Mass Flux into and out of an incremental volume

2.1 AN EXAMPLE IN CONTINUITY

Under conditions of mass conservation, an equation of continuity may be derived for the generalized conditions of compressible fluid motion in three directions. In cartesian coordinates, the control volume is useful to derive an expression for the conservation of mass. As indicated in Figure (1), the control volume ΔV is composed of three axes whose product defines the size of the differential volume or cube (i.e. $\Delta V = \Delta x \Delta y \Delta z$). The total mass flux ($n_{tx}$) across the x face of the cube is then simply the

product of the bulk density of the fluid (ρ) and the velocity of flow in that direction (v_x). For all six faces of the control volume, the mass balance becomes:

$$\Delta_x(n_{tx}\Delta A_x) + \Delta_y(n_{ty}\Delta A_y) + \Delta_z(n_{tz}\Delta A_z) - \Delta(\rho\Delta V)/\Delta t = 0 \qquad (1)$$

where

n_{ti} = total mass flux of the soil water in the ith direction (i = x,y,z),

ρ = fluid density,

Δt = time increment,

$\Delta A_X = \Delta z\Delta y$, $\Delta A_Y = \Delta z\Delta x$, $\Delta A_Z = \Delta x\Delta y$,

$\Delta_i(n_{ti}\Delta A_i) = (n_{ti}\Delta A_i)\,|_{i+\Delta i} - (n_{ti}\Delta A_i)\,|_i$ = net mass flow of the soil water through the incremental area ΔA_i, and

$\Delta(\rho\Delta V)/\Delta t$ = bulk accumulation term

Dividing Equation (1) by ΔV, realizing that soil water is an incompressible fluid under ambient conditions ($\Delta(\rho\Delta V)/\Delta t \rightarrow 0$), and taking the limits as $\Delta x \rightarrow 0$, $\Delta y \rightarrow 0$, $\Delta z \rightarrow 0$ yields:

$$\partial n_{tx}/\partial x + \partial n_{ty}/\partial y + \partial n_{tz}/\partial z - \partial\rho/\partial t = 0 \qquad (2)$$

Letting n_{ti}, the total mass flux, equal the product of the fluid density (ρ) and the fluid velocity in the ith flow direction (v_i):

$$n_{ti} = \rho v_i \qquad (3)$$

and Equation (2) becomes:

$$\rho\,(\partial v_x/\partial x + \partial v_y/\partial y + \partial v_z/\partial z) - \partial\rho/\partial t = 0 \quad \text{or}$$

$$\nabla \cdot \rho\mathbf{v} - \partial\rho/\partial t = 0 \qquad (4)$$

where $\mathbf{v} = v(x,y,x)$.

For the special case of constant fluid density through time, $\partial\rho/\partial t \rightarrow 0$, and Equation (4) simplifies to:

$$\partial v_x/\partial x + \partial v_y/\partial y + \partial v_z/\partial z = 0 \quad \text{or}$$

$$\nabla \cdot \mathbf{v} = 0 \qquad (5)$$

Equations (4) and (5) represent the equations of continuity for the simple control volume of Figure (1) under the generalized conditions of incompressible flow. An equivalent symbolic notation for this continuity is provided in Figure (2).

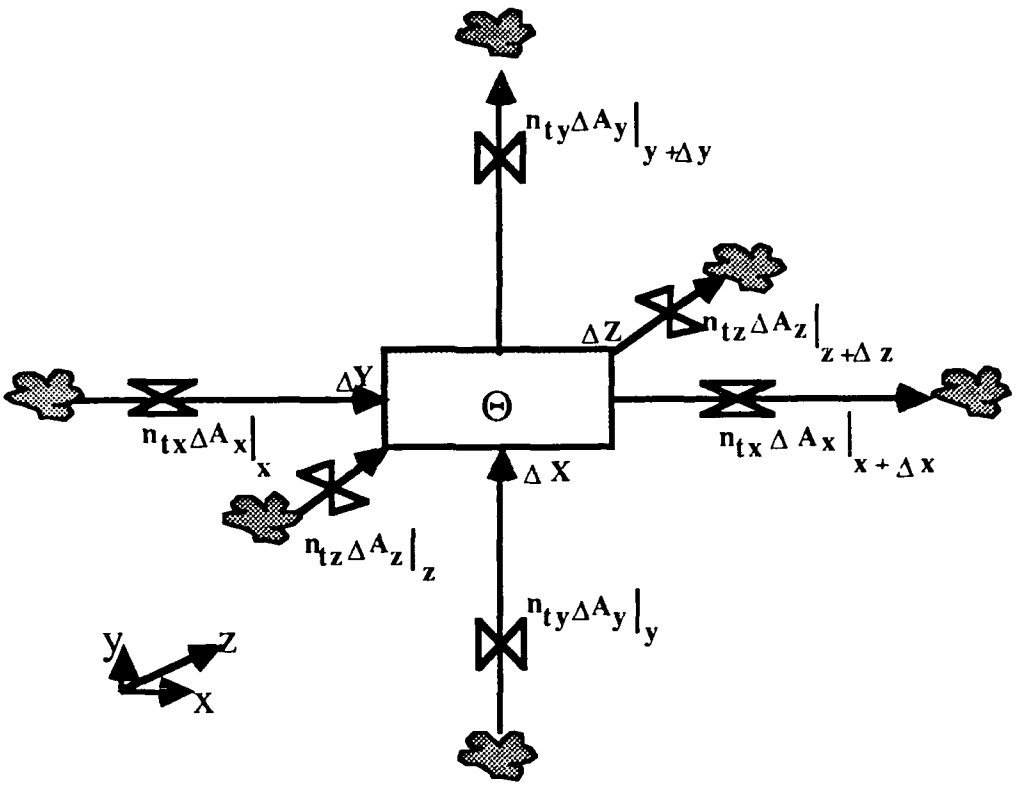


Figure 2. Systems representation for mass flux into and out of an incremental volume

The notation is a follows[2] :

(1) **Clouds** represent *Sources or Sinks.* If an arrow points into the cloud it must be a sink. Conversely, an arrow pointing away from a cloud implies that the cloud must be a source. In Figure (1), the cloud associated with precipitation is a source, the cloud associated with evapotranspiration may be either a source or a sink depending on the physical process (e.g. a source during periods of evaporation, a sink during periods of condensation).
(2) **X-valves** represent *Rates (differentials).* The object is meant to symbolize a "plumber's" valve that opens or closes depending on physical conditions. In the precipitation case, the valve will open during periods of rainfall and water will "flow" from the cloud (the source) into the rectangle.

[2] In addition, variations on the arrows used to connect sources, sinks, and levels are possible depending on the physical system. For example, "dashed" or "light-faced" arrows are used to infer correlation and information relationships.

(3) **Rectangles** represent *Levels (integral equations)*. Levels accumulate or deplete depending on the X-valves that are connected to them (i.e. they are assigned an initial condition, and then allowed to integrate the differential equations symbolized by the rates). The rectangles are also referred to as the "State Variables" for the system since they have the capacity to change *states* through time and space. The term "steady-state" is used to describe a state variable invariant in time (and/or space).

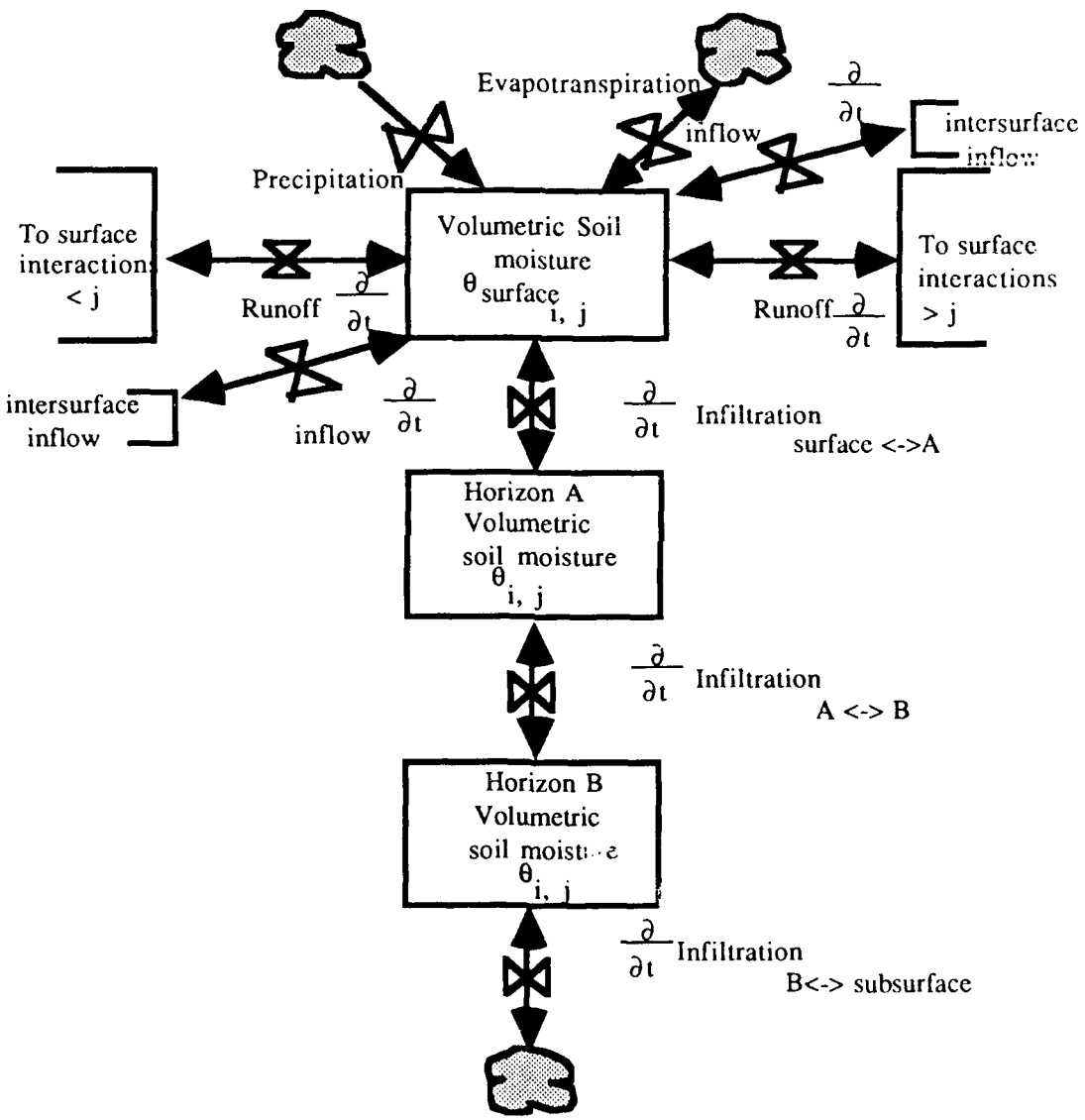


Figure 3. Volumetric soil moisture balance using state-space notation. Intersurface flow between pixels is shown in the lower-right to upper-left diagonal rate components and not displayed for a 360 rotation to simplify the diagram. Notation "<->" indicates direction of interaction (e.g. A<->B indicates interaction between Horizon A and Horizon B for differential Infiltration)

In the systems representation of Figure (2), mass enters and exits the control volume (labeled Θ) according to the mass rates of flow into and out of the rectangle. Mass originates and exits the system boundary via sources and sinks (Clouds). The mass is stored at any point in time within the Rectangle (Integral Equation) according to the rates of flow (Σ-valves) that are used to direct mass into and out of the rectangle.

The notation may be extended using rates to symbolize complex physical processes. In this capacity, rates symbolize sub-sectors or sub-aggregates, and are used to differentiate the otherwise complex flows into simple constituent elements (atoms). An example of the process is shown in Figure (3), using a primary state variable Θ to symbolize the random vector for volumetric soil moisture.[3] The figure indicates that the major mass transfer processes have been divided according to constituent rates of flow for: Precipitation, Evapotranspiration, Infiltration, Runoff, and Intersurface Flow. Each of these rate processes represents a major sector in the model whose behavior is defined according to the terrain and meteorological conditions.[4]

Following the flow of moisture in Figure (3), precipitation enters the system via a source component (Cloud) whose rate is determined by real-time meteorological data. At any point in time, moisture may accumulate or decrement depending on the mass rate of flow from the rate processes acting on the storage of moisture (rectangle $\Theta_{i,j}$). During periods of evaporation, moisture leaves $\Theta_{i,j}$, and returns during periods of condensation (double sided arrow). Both runoff and inflow are surface effects that flow between adjacent surface pixels. Infiltration routes moisture into the sub-surface regime, and mass can flow up, out of the lower soil horizons, during periods of reverse-infiltration and hydrostatic flow.

In sum, the symbolic notation is a robust method for displaying complex systems of differential equations. By explicitly representing sources, sinks, integral, and differential equations, rate processes may be displayed in their constituent elements. Both sources and sinks are easily recognized within the system diagram, and arrows represent the direction of flow between operators. Other operators, such as auxiliary equations,

---

[3]Bold scripting is used to reinforce the vector convention: the behavior of Θ varies in time and space with functional notation $\Theta \rightarrow \Theta(x,y,z,t)$ in gridded rectangular coordinates.

[4]Figure (3) represents a "skeleton" diagram and illustrates a few of the major forces that affect volumetric moisture content by position (the i,j subscripting) and by representative soil horizon (Surface, Horizon A, Horizon B). As indicated, rates have been aggregated into processes (Precipitation, Evapotranspiration, Infiltration, Intersurface Flow, and Runoff) to simplify the presentation. Actual rates are determined using terrain, meteorological, spatial, and temporal factors. For example, the model in Appendix 1 uses Soil Conservation Service algorithms to determine evapotranspiration, runoff, and infiltration. Since each model is a "plug-in" unit, many algorithms can be easily applied to a rate process for purposes of validatation and simulation.

and information arrows are added to the notation to display the finer details of the state-space model.[5]

3. SPATIAL AVERAGING

The state-space method can be applied to problems in spatial averaging by creating a network (or mesh) of state variables that corresponds to the type of primary data available (i.e. format of the real-time data: grid cell or polygon). In a gridded format, soil moisture states are displayed using standard matrix notation (i,j) in multiple soil horizons. For polygon data, minimum rectangles may be used to "rasterize" the data or centroids selected to subscript the polygons by location (e.g. centoid location). Each pixel corresponds to a single volumetric moisture state Θ, and overlays are created to depict moisture states in multiple soil horizons.[6] Therefore, Θ becomes a function of its position within the matrix (e.g the region of the watershed), its soil horizon position (H), and its prior behavior through time (t).

Under a spatial averaging scenario, Θ is a function of neighboring pixels. Hence, the behavior of surrounding states influences the behavior of the modeled state (i.e an autodynamic autospatial process).[7] For the routing of volumetric soil moisture, probability assignments may be weighted according to saturation conditions: during periods of saturation by surrounding pixels, there is a higher probability of surface or groundwater flow. Conversely, periods of draught may force a higher infiltration and absorption capacity that detracts from surface and groundwater flow.

In the following sections, a linear averaging algorithm is developed using a generalized grid-cell format. Extensions to polygon data are discussed, and a heuristic nine-pixel algorithm is developed. The model organization is overviewed and heuristics are provided to illustrate an operating sequence for the model.[8]

---

[5]The state-space diagrams in Figure(2) and Figure(3) show simple "skeletal" structure. Both auxiliary equations and information arrows have been removed to simplify the presentation. Auxiliary equations may be in the form of an algebraic equation of in interpolated graphical form. Information arrows are used to handle the non-physical flows of the system (e.g. correlation between elements in the models).

[6]Pixels are assigned grey scale weights (or color indices) depending on the corresponding volumetric moisture content Θ by (i,j) position within the gridded file. The minimum grid cell size and the iteration interval are selected according to user definitions depending on the availability and format of real-time digital information.

[7] Neighboring information becomes extremely important in the estimation of volumetric moisture content under sparse data conditions. Under these constraints, neighboring information is typically weighted according to the "inverse-square" of the pixel distance from the state location.

[8] One extension is to spatially average large images using weather radar and multispectral data from Landsat and SPOT. In this approach, spectral signitures are used as initial conditions for the state-space algorithm, and images are spatially averaged using models that link the hydrodynamic budget to the gray-scale range for a particular spectral band.

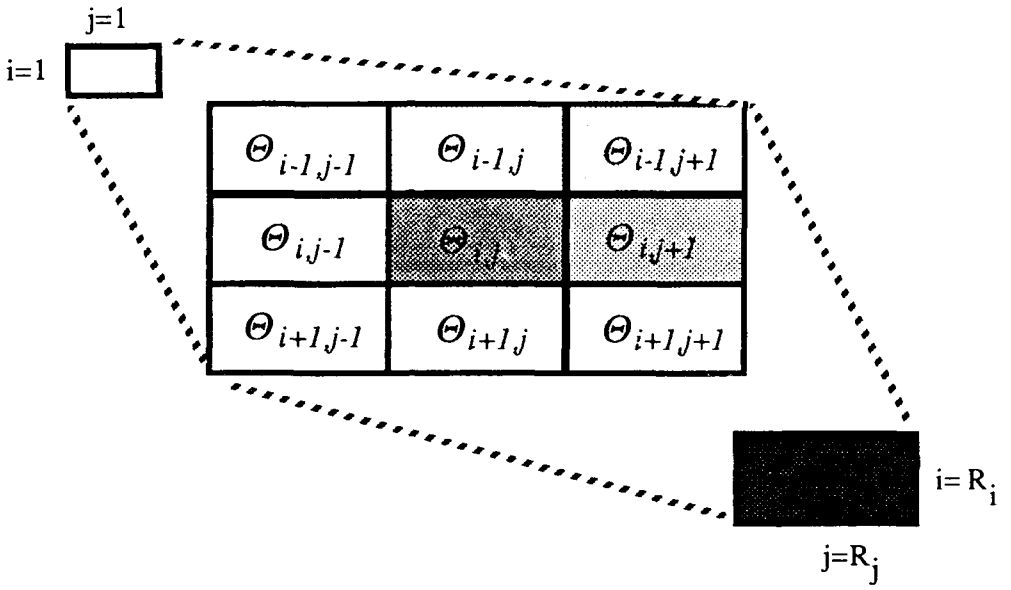


Figure (4) : An index notation to define a grid cell spatial average. Each pixel corresponds to a moisture state and is displayed in grey-scale (color) corresponding to the percent volumetric moisture for that state-space location. R_i and R_j define the lower right corner of the test region

3.1 LINEAR AVERAGING ALGORITHM

In a linear spatial averaging algorithm, neighboring pixels are assigned weights according to a priori criteria (e.g. location, land use, elevation, soil type), and summed to produce a weighted average of the state variables. An example of this process is provided in Figure (4). In this case the following event sequence defines the spatial average:

(1) A weighting criteria (ω) is developed (typically based on terrain factors or state location to known data, such as inverse square weighting).

(2) A criteria is outlined for handling conditions outside the test region. In Figure(4), gridded data is displayed with (i,j) subscripts. Hence averaging criteria must be established for boundary cases where:

$$\text{(a) } \{i < 1\}, \text{(b) } \{j < 1\}, \text{(c) } \{i > R_i\}, \text{(d) } \{j > R_j\} \qquad (6)$$

and R_i and R_j are the respective row and column bounds that define the grid-cell region (composed of R_i x R_j states).

For polygon data, Figure (5), the enclosing region R is such that all polygons in the spatial average are contained within the perimeter of R. If (i,j) subscripts are assigned to the centroid of each polygon Θ_i, then conditions established for gridded data may be modified to polygon data.

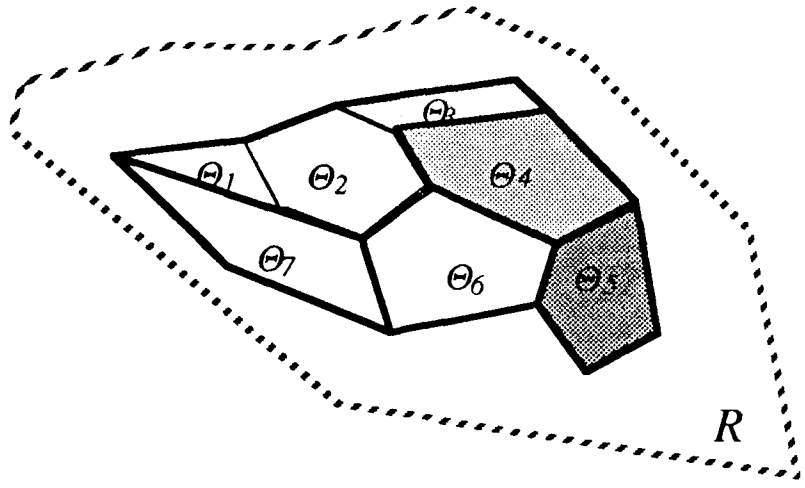


Figure (5): A polygon representation of the volumetric moisture states within the total region R. Polygons may be averaged using centroid positions for each Θ_k polygon

(3) An iteration scheme is developed. In gridded data, a double loop is commonly used to step between adjacent states (e.g While $j < R_j$ do, While $i < R_i$ do....), and

(4) A spatial window is developed to determine the number of pixels to use in the average. In a composite nine pixel scheme[9], the number of pixels per average is bounded by the location of the state variable($\Theta_{i,j}$) in the matrix:

if (i = 1 & j =1) then
$$\Theta_{i,j} = 1/4\Sigma\omega_k\,(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j+1}+\omega_3\Theta_{i+1,j}+\omega_4\Theta_{i+1,j+1}); \quad (7a)$$

else if (i=1 & j =R_j) then
$$\Theta_{i,j} = 1/4\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j-1}+\omega_3\Theta_{i+1,j}+\omega_4\Theta_{i+1,j-1}); \quad (7b)$$

else if (i=1 & j <>1 & j <>R_j) then
$$\Theta_{i,j} = 1/6\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j-1}+\omega_3\Theta_{i,j+1}+\omega_4\Theta_{i+1,j} +\omega_5\Theta_{i+1,j-1}+\omega_6\Theta_{i+1,j+1}); \quad (7c)$$

else if (i=R_i & j =1) then
$$\Theta_{i,j} = 1/4\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j+1}+\omega_3\Theta_{i-1,j}+\omega_4\Theta_{i-1,j+1}); \quad (7d)$$

else if (i=R_i & j =R_j) then
$$\Theta_{i,j} = 1/4\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j-1}+\omega_3\Theta_{i-1,j}+\omega_4\Theta_{i-1,j-1}); \quad (7e)$$

[9]A composite nine pixel scheme implies that a maximum of nine pixels are included in the spatial average, and as few as one pixel may be used depending on boundary conditions. In the outline algorithm:

(1) four pixels are used at corner points that define the region,
(2) six pixels are used for pixels with rows (columns) that border the regional boundary
(3) nine pixels are used for all mid- row (column) pixels.

else if ($i=R_i$ & $j<>1$ & $j<>R_j$) then

$$\Theta_{i,j} = 1/6\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j-1}+\omega_3\Theta_{i,j+1}+\omega_4\Theta_{i-1,j}+\omega_5\Theta_{i-1,j+1}+\omega_6\Theta_{i-1,j-1}); \quad (7f)$$

else if ($j = 1$ & $i > 1$ & $i < R_i$) then

$$\Theta_{i,j} = 1/6\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j+1}+\omega_3\Theta_{i-1,j}+\omega_4\Theta_{i+1,j}+\omega_5\Theta_{i+1,j+1}+\omega_6\Theta_{i-1,j+1}); \quad (7g)$$

else if ($j = R_j$ & $i > 1$ & $i < R_i$) then

$$\Theta_{i,j} = 1/6\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j-1}+\omega_3\Theta_{i+1,j}+\omega_4\Theta_{i-1,j}+\omega_5\Theta_{i-1,j-1}+\omega_6\Theta_{i+1,j-1}); \quad (7h)$$

else if ($j > 1$ & $j < R_j$ & $i > 1$ & $i < R_i$) then

$$\Theta_{i,j} = 1/9\Sigma\omega_k(\omega_1\Theta_{i,j} +\omega_2\Theta_{i,j-1}+\omega_3\Theta_{i,j+1}+\omega_4\Theta_{i-1,j}+\omega_5\Theta_{i+1,j}+\omega_6\Theta_{i+1,j+1} +\omega_7\Theta_{i-1,j-1}+\omega_8\Theta_{i+1,j-1}+\omega_9\Theta_{i-1,j+1}); \quad (7i)$$

where

ω_k = the assigned weight of the kth pixel.

In the spatial averaging model provided in Appendix (1), a composite nine pixel scheme is provided with a number of refinements:

(1) Pixel weights are derived from: (a) normalized elevation gradients that produce maximum likelihood routing directions, and (b) normalized radial proximity measures that adjust the weights (ω) to their proximity (r.e. the base state pixel being spatially averaged).
(2) Pixels are spatially averaged and then numerically integrated into the hydrodynamic model.
(3) Pixels are only spatially averaged if the hydrologic budget determines that water is available for re-routing.
(4) Pixels are spatially averaged in three dimensions: averaged first for surface effects, then re-averaged by soil horizon based on fractional infiltration rates.
(5) The composite nine pixel scheme is flexible. If differentials across pixel boundaries are few, the averaging window will shrink to a minimum size of one pixel. Large discrepancies between distant pixels expands the averaging window to a maximum size of the full region. The window expands and contracts in a geometric progression proportional to the hydrodynamic activity in the region (i.e. 1,4,9,16,25,36....).

A detailed listing of the averaging method is provided in Appendix (1) under the section: UNIT Spatial. Three major procedures accomplish the averaging process: (1) SpaceAvg—used to spatially average surface and subsurface affects within a soil horizon, (2) DepthAvg—used to depth average between soil horizons, and (3) ThetaFlux—used to calculate the hydrologic budget to determine the volume of water available for re-routing.

3.2 ORGANIZATION

The model in Appendix (1) is organized according to the major processes required to : (i) Initialize and "launch" the system, (ii) perform a hydrodynamic budget by state location, (iii) spatial average the water available for re-routing, and (iv) display the results. The computer code is organized into constituent UNITS that, acting together, perform these major tasks. In outlining the operating sequence of the model, ***bold Italics*** is used to designate sections within the code were readers can locate the detailed information. For example, runoff information is located in the "UNIT Runoff" section of Appendix (1). In referring to this section, the script notation "***Runoff***" (rather than "UNIT Runoff") will be used.

3.2.1 Initialization and launch

The system is initialized and launched in the section ***Initialize*** described in Appendix (1). Initial and Boundary Conditions are summarized by variable name in Table (1). Values that correspond to Soil Conservation Service units are listed in parenthesis as "{SCS}".

Table 1. Primary Variables used by the Spatial Averaging Model

Variable	*Value*	*Range*	*Structure*
Soil Type {SCS}	CoarSand = 1; CoarSLoam = 2;	1 to 16	Array by i,j of Integer
Soil[i,j]	Sand = 3; LoamSand = 4; LoamFineSand = 5; SandLoam = 6; FSandLoam = 7; VFSandLoam = 8; Loam = 9; SiltLoam = 10; SanClayLoam = 11; ClayLoam = 12; SilClayLoam = 13; SandClay = 14; SilClay = 15; Clay = 16;		
Soil Horizons {SCS} *SoilDepth*	Surface = 1; AHorizon = 2; BHorizon = 3;	1 to 3 {0-3"} {3-6"} {3-6"}	Integer
Soil Infiltration Index {SCS} *SoilInfil*	LowRun = 1; ModInfil = 2; SlowInfil = 3; HighRun = 4;	1 to 4	Integer **or** Array by i,j of Integer

Table 1 (cont'd). Primary Variables used by the Spatial Averaging Model

Variable	*Value*	*Range*	*Structure*
Soil Treatment Index {SCS} *SoilTreat*	StrRow = 1; Contoured = 2; ContTerrace = 3;	1 to 3	Integer **or** Array by i,j of Integer
Soil Hydrologic Condition {SCS} *SoilCond*	Poor = 1; Fair = 2; Good = 3;	1 to 3	Integer **or** Array by i,j of Integer
Growth Index *GI* {SCS}	%	0 to 100	Integer
Land Use Class {SCS} *LandUse[i,j]*	Fallow = 1; RowCrop = 2; SmGrain = 3; Legume = 4; Pasture = 5; Meadow = 6; Woods = 7; Farm = 8; DirtRoad = 9; HardRoad = 10;	1 to 10	Array by i,j of Integer
Elevation *Elev[i,j]*	ft or m	> 0	Array by i,j of Real
WindSpeed *Wind*	mph or m/sec	> 0	Real **or** Array by i,j of Real
Temperature *Temp*	degr F or degr C	ambient	Real **or** Array by i,j of Real
Relative Humidity *Humid*	% RH	0 to 100	Real **or** Array by i,j of Real
Granularity		> 0	Real
dx	km	1 km {base}	Integer
dy	km	1 km {base}	Integer
dt	hr or day	1 hr {base }	Integer

3.2.2 Hydrodynamic budget

The hydrodynamic budget is calculated within the procedure "ThetaFlux" in the ***Spatial*** section of Appendix (1). This unit "calls" other units within the code to determine the differential: (i) ***Runoff***, (ii) ***Evapotranspiration***, (iii) ***Infiltration***, (iv) ***Precipitation***, (v) Inflow, and (vi) Outflow by (i,j) location. The ThetaFlux procedure calculates the differentials by calling the other units, and combines their effects to pro-

duce a "net" differential, used to numerically integrate the result (based on previously averaged states).

3.2.3 Inclusion of the hydrodynamic budget

The hydrodynamic budget is numerically integrated using a Spatial Averaged prior state for volumetric moisture content. In Eulers form, the following operations are completed:

$$\Theta_{H,i,j,t} = \text{Spatial Average}(\Theta_{H,i,j,t-1}) + dt\ (d\Theta_{H,i,j,t}/dt) \quad (8)$$

where

$\Theta_{H,i,j,t}$ = The volumetric moisture content at surface position i,j for soil horizon H at time t,

Spatial Average($\Theta_{H,i,j,t-1}$) = the spatially average prior state (time t-1) at surface position (i,j) for soil horizon H (a composite nine pixel spatial average operator is defined in equations (7a) through (7i)),

dt = the differential time step, and

$d\Theta_{H,i,j,t}/dt$ = d/dt(Precip) - d/dt(Runoff) - d/dt(Evapotranspiration) - d/dt(Infiltration) + d/dt(Inflow) - d/dt(Outflow) = the "net" differential for surface position (i,j) at time t, for horizon H.

In the computer code of Appendix (1), the spatial averaging process is split into two separate operators that act in conjunction to perform spatial averaging: (1) with soil horizons (SpaceAvg), and (2) between soil horizons (DepthAvg). Details of each process are provided in ***Spatial***.

3.2.4 Display

Two and three dimensional displays are currently supported depending on the dimensionality of the simulation. Example simulations and discussion are provided in Appendix (2).

3.3 SEQUENCE

For the model in Appendix (1), water is routed and spatially averaged in the following computational sequence:

time t=0:

(1) State, terrain, and meteorological variables, are assigned initial conditions.
(2) Elevation profiles are normalized and gradients determined. The gradients are used to determine maximum likelihood routings of soil moisture subject to terrain conditions.

repeat (t > 0)

for each surface pixel do:
begin

(1) Calculate Precipitation from real-time meteorological data

(2) Determine runoff as a function of:
(i) Free Drainage,
(ii) Surface Gradient,
(iii) Landuse,
(iv) Soil treatment (compaction, terracing et. al)
(v) Soil condition
(vi) Precipitation

(3) Determine Evapotranspiration as a function of:
(i) Soil Type,
(ii) Landuse,
(iii) Wind speed,
(iv) Temperature
(v) Relative Humidity
(vi) Free Drainage

(4) Determine Surface Inflow and Outflow as a function of terrain and meteorologic data

for each soil horizon do
begin

(5) Determine Infiltration as a function of:
(i) Permeability by Soil Type,
(ii) Soil Type,
(iii) Landuse,
(iv) Soil condition
(v) Surface Gradient

(6) Calculate the total differential for the mass balance in volumetric units:
$d\Theta_{H,i,j,t}/dt$ = d/dt(Precip) - d/dt(Runoff) - d/dt(Evapotranspiration) - d/dt(Infiltration) + d/dt(Inflow) - d/dt(Outflow)

(7) Calculate a new volumetric moisture state based upon an a priori spatial average of the volumetric moisture state and the calculated total differential:

$$\Theta_{H,i,j,t} = \text{Spatial Average}(\Theta_{H,i,j,t-1}) + dt\ (d\Theta_{H,i,j,t}/dt)$$

(8) Extract the real-time state data Θ^R. If is significantly different than the estimated state Θ, then adjust estimate and re-spatial average the adjusted value:[10]

$$\text{if } |(\Theta - \Theta^R)| > \delta \text{ then } \Theta = \Theta^R +/- \varepsilon \qquad (9)$$

where
δ = the tolerance criteria, and
ε = the error adjustment

end; {for each soil horizon do}
end; {for each surface pixel do}

until end of simulation

4. CONCLUSIONS

State-space theory is used to spatially average soil moisture in two and three dimensions using new symbolic techniques to build systems of differential equations. The process allows for the modeling of key hydrodynamic elements required to define the primary mechanics of flow. The hydrodynamic budget is combined with a spatial averaging shell to produce a method for estimating soil moisture using weighted information from nearest neighbors and weighted information throughout the test region. The model uses a flexible window format for spatially averaging soil moisture within horizons, and separates two and three dimensional flow using independent depth averaging methods based on infiltration properties of the soil. Preliminary results, Appendix (2), indicate that the method is computationally efficient and produces "intuitive" routings under unusual flow conditions.[11]

Future efforts include a calibration and accuracy assessment for the model and a refinement of the algorithm used for real-time data adjustment. The method will be tested and validatedusing observed real-time data from two Iowa watersheds.

---

[10]It is implicitly assumed that deviations are due to estimation error rather that measurement or calibration error.

[11]The model runs under 512k of memory on an Apple Macintosh and has been written in Vanilla Pascal for easy portability to other operating environments. An iteration of the model (10 x 10) display) requires less than ten seconds on a Macintosh II.

REFERENCES

Anderson, B. D. O., and J. B. Moore, *Optimal Filtering*, Prentice-Hall, EngleWood Cliffs, N.J., 1979.

Burgess, T.M.,Webster, R., and A. B. McBratney, "Optimal Interpolation and Isarithmic Mapping of Soil Properties: 4 Sampling Strategy.," *J. Soil Sci.*, 32, 1981.

Chirlin, G. R., and E. F. Wood, "On the Relationship Between Kriging and State Estimation", *Water Resour. Res.*,18, 2, 1982.

Clark, I., *Practical Geostatistics*, App. Sci. Pub., London, 1982.

Forrester, J. W., *Principles of Systems*, MIT Press, Cambridge, Mass., 1968.

Gelb, A. (Ed.) *Applied Optimal Estimation*, MIT Press, Cambridge, Mass, 1974.

Gelhar, L. W., and C. Axeness, "Three-dimensional Stochastic Analysis of Macrodispersion in Aquifers," *Water Resour. Res.*, 19, 1983.

Journel, A. G., and C. J. Huijbregts, *Mining Geostatistics*, Academic Press, London, 1978.

Kitanidis, P. K.,"Statistical Estimation of Polynomial Generalized Covariance Functions and Hydrologic Applications," *Water Resour. Res.*, 19, 1983.

Krige, D.G., "Two Dimensional Weighted Moving Average Trend Surfaces For Ore Valuation," *Journal of the South African Institute of Mining and Metallurgy*, March, 1966.

LaPotin, P., McKim, H.L., and T. Pangburn, "An Analysis of Variance in Soil Moisture Measurement Techniques : A Comparison of Radio Frequency, Gravimetric, and Tensiometric Measures," CRREL Special Report (in prep), Hanover, N.H., 1987.

Matheron, G., "Principles of Geostatistics," *Economic Geology*, n8, v58, 1246–1266, Dec., 1963.

Matheron, G., "Les Variables Regionalisees et leur Estimation," Masson, Paris, 1965.

Matheron, G., "La Theorie des Variables Regionalisees et ses Applications," Ecole Nationale Superieure des Mines de Paris, *Cah. Cent. Morphol. Math. Fontyainebleau*, n5, 1-212, 1970.

McBratney, A. B., and R. Webster, "The Design of Optimal Sampling Schemes for Local Estimation and Mapping of Regionalized Variables : 2, *Comput. Geosci*, 7, 1981.

McBratney, A. B., and R. Webster, "How Many Observations are Needed For Regional Estimation of Soil Properties?," *Soil Science*, V135, N3, March 1983.

McKim, H., and P. LaPotin, "Spatial Averaging of Soil Moisture," CRREL Special Report (in prep), Hanover, N.H., 1987.

Olea, R.A., "Optimal Contour Mapping Using Universal Kriging," *J. Geo. Research*, 1974.

Papoulis, A., *Probability, Random Variables, and Stochastic Processes*, McGraw-Hill, New York, 1965.

Russo, D., and W. A. Jury, "A Theoretical Study of the Estimation of the Correlation Scale in Spatially Variable Fields, 1. Stationary Fields," *Water Resources Research*, V23, N7, 1257–1268, July, 1987a.

Russo, D., and W. A. Jury, "A Theoretical Study of the Estimation of the Correlation Scale in Spatially Variable Fields, 2. Nonstationary Fields," *Water Resources Research*, V23, N7, 1257–1268, July, 1987b.

Webster, R., and T. M. Burgess, "Optimal Interpolation and Isarithmic Mapping of Soil Properties: 3. Changing drift and Universal Kriging," *J. Soil. Sci.*, 31, 1980.

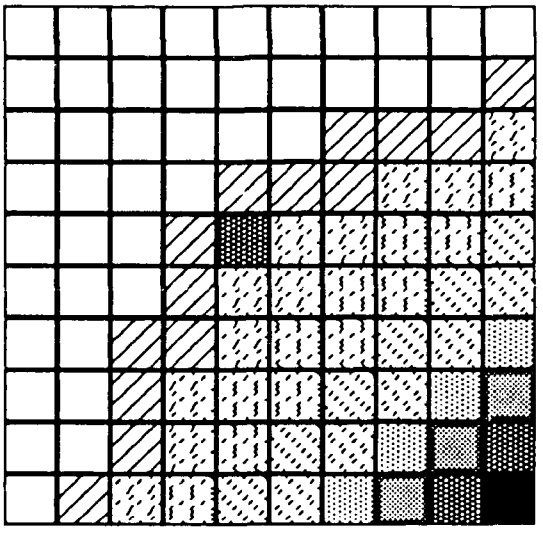
Yeh, W. W-G., and Y.S Yoon, Aquifer Parameter Identification with optimum Dimension in Parameterization, *Water Resource Research*, 17(3), 664-672, 1981.

APPENDIX 1: VANILLA PASCAL CODE FOR THE SPATIAL AVERAGING MODEL*

* [Editor's Note: Space limitations preclude printing the 26 pages of program listings in these proceedings. The Cold Regions Research Engineering Laboratory should be contacted to obtain the program listing.]

APPENDIX 2. PRELIMINARY RESULTS

A prototype dynamic spatial averaging *shell* was constructed to begin feasibility testing of the state-space method for two and three dimensional flow using simulated real-time gridded data. Each simulation assumes the following Soil Conservation Service (SCS) conditions: (1) controlled soil treatment, (2) fair soil conditions, (3) sandy loam soil type, and (4) wooded land use. In addition, the following meteorological/ climatic and insitu sensor conditions were considered: (1) 1 mph wind speed, (2) temperature 50° F, (3) 10% relative humidity, and (4) an initial volumetric soil moisture content of 20% [12]. These conditions were held constant (i.e. steady-state meteorological conditions) so that we could clearly illustrate the combined effect of *only* precipitation and elevation on the spatial averaging of soil moisture. In each array pixels are displayed as variable grey-scale intensities depending on the volumetric moisture value by location within the array. Grey-scales were assigned intensity from white to black depending on the volumetric soil moisture value, which could vary from 0 to 100%, respectively. For elevation plots, grey-scales were assigned depending on elevation contours (i.e. darker textures indicate higher elevations and white indicates the minimum



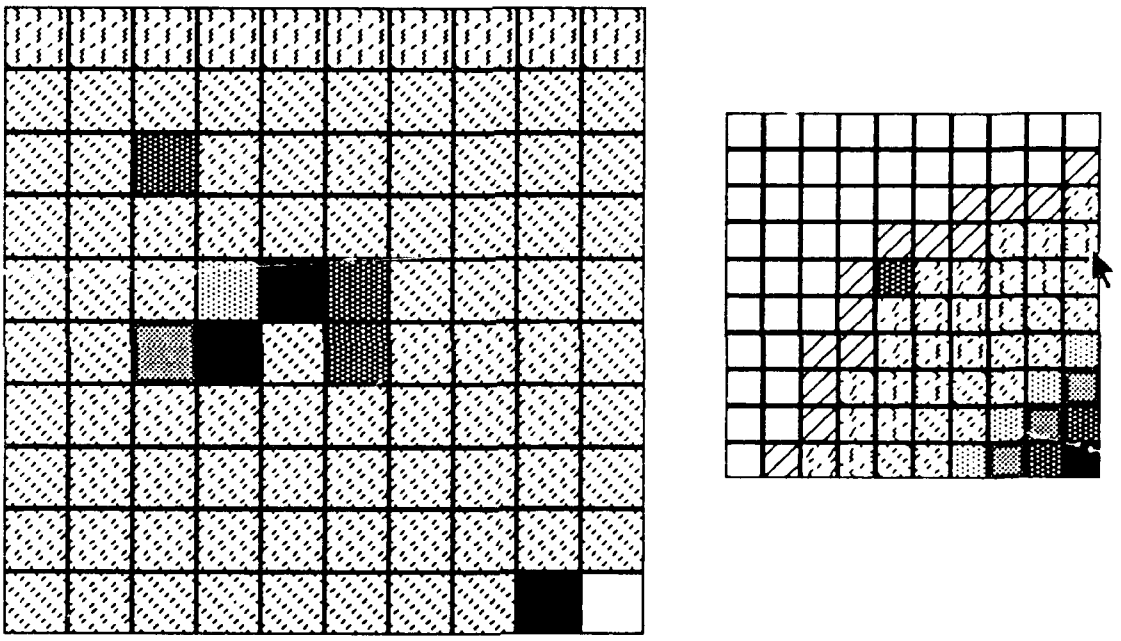
Appendix Figure 1. Elevation profiles for the spatial average simulation

[12] A detailed listing of boundary and initial conditions may be found in McKim and LaPotin [1987].

elevation in the region). All grids are displayed on a 1 km by 1 km basis within a 10km by 10km test region. In each simulation, states are re-computed and real-time data is integrated into the system on an hourly basis.

ELEVATION

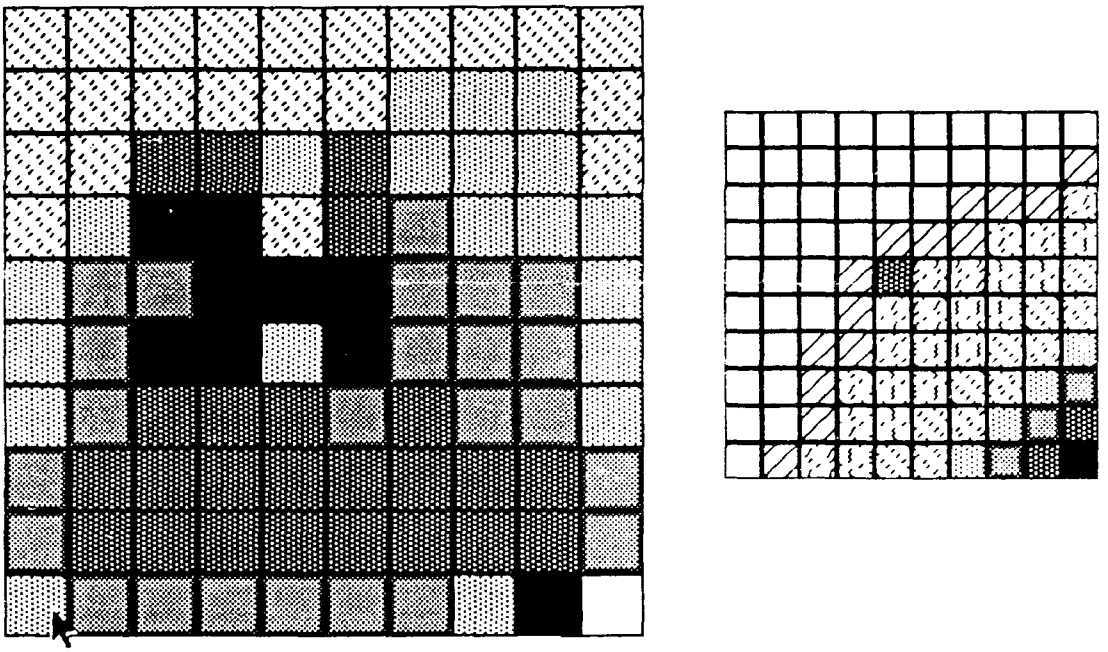
The elevation profile for the region is provided in Appendix Figure (1). Initial conditions indicate a "mole-hill" at position (5,5) within the 10 km by 10 km matrix, and an inner basin between the mole-hill and the ridge at position (10,10). Otherwise, the elevation level is gradually declining from lower-right to upper-left.



Appe ix Figure 2. Two hours into the base simulation. Volumetric soil moisture profile is shown to the left; elevation profile is shown to the right

TWO DIMENSIONAL SIMULATION

In Appendix Figures (2) through (4), two dimensional routings of volumetric soil moisture content are provided. For each simulation, moisture influx at position (5,5), the centroid of the "mole-hill", and at position (10,9) adjacent to the crest of the elevation profile is given as a *saturated* steady-state boundary condition. Darker textures indicate greater degrees of saturation. Precipitation conditions dictate saturated flow at positions



Appendix Figure 3. Four hours into the base simulation. Volumetric soil moisture profile is shown to the left; elevation profile is shown to the right

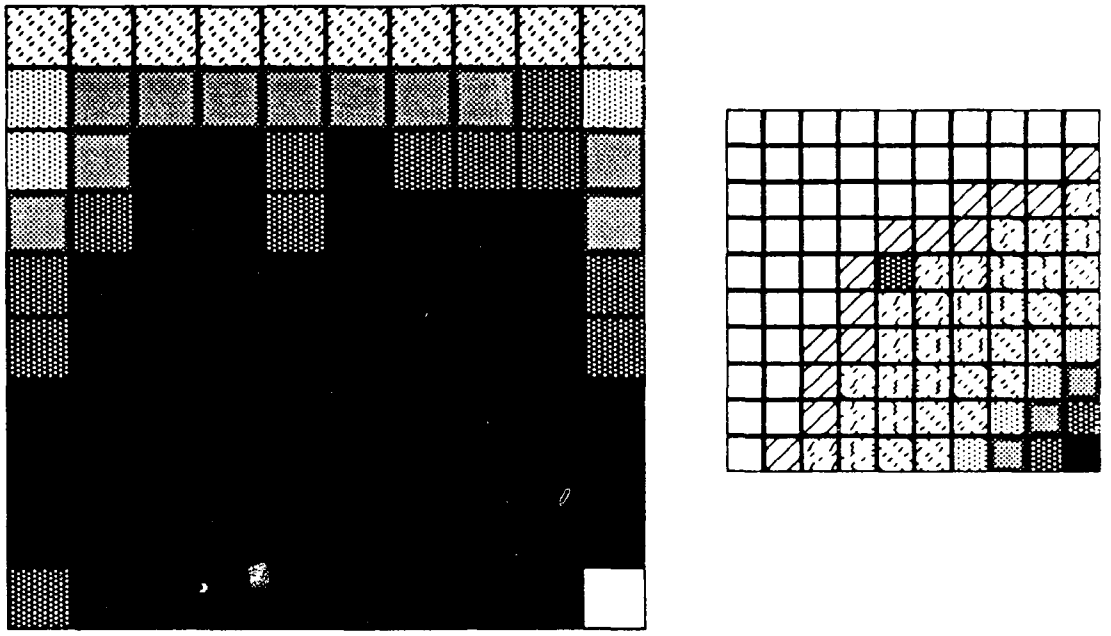
(5,5) and (10,9). As an example of using real-time insitu data, the CRREL/ Dartmouth radio frequency probe[13] indicates that near saturated conditions (40% volumetric moisture) are present at position (3,3). Moisture is being routed away (northwest) from the point of maximum elevation at position (10,10) and down the gradient. After two hours, complex routing is beginning to develop near the "mole-hill" shown in Figure (6).

Four hours into the simulation, the saturated flow conditions at positions (5,5) and (10,9) have produced near saturated flow regimes throughout the inner-basin between the "mole-hill" and point of maximum elevation in the lower right of the 10 km by 10 km matrix (Figure (7)). Saturated flow at position (5,5) is being fully routed, via overland flow, away from the "mole-hill" and saturating the basin located in the upper left (i.e. positions (4,3), (4,4), (5,4)). Water is routed according to the degree of saturation and the elevation gradients within the grid array.

Six hours into the simulation the inner basin is completely saturated and marginal areas are approaching saturation (Appendix Figure (4)). The "over-flow" of precipitation at positions (5,5) and (10,9) is being routed from lower right to upper left according to the regional topographic gradient. Conditions appear saturated due to the combined effects of poor infiltration of surface water into the silt clay loam soil, and very large rates of precipitation that far exceed the rates of runoff and evapotranspiration.

---

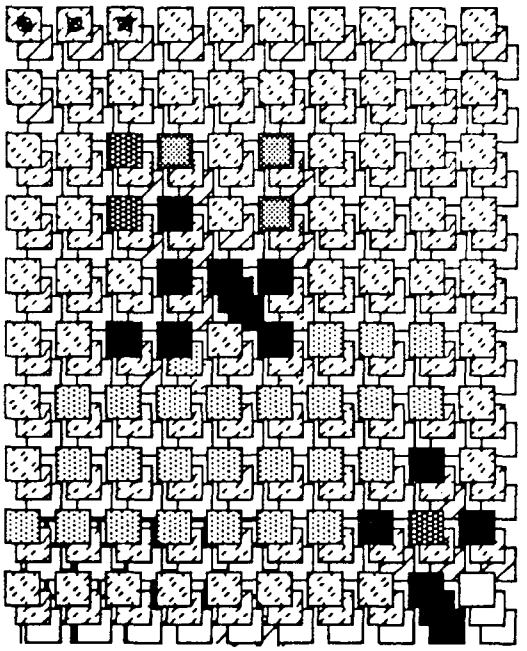
[13] re. LaPotin et al. [1987].



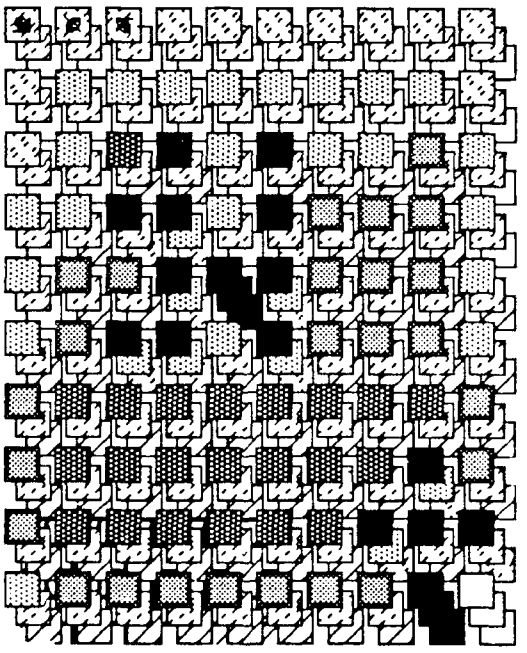
Appendix Figure 4. Six hours into the base simulation. Volumetric soil moisture profile is shown to the left; elevation profile is shown to the right

THREE DIMENSIONAL SIMULATION

In Figures (9) and (10), the two dimensional routing is extended into a three dimensional regime (three soil horizons are displayed using off-set pixels by location). The simulation uses three specific Soil Conservation Service soil types for horizons of known depth: (1) sand clay @ 0 to 3 inches, (2) sand clay loam @ 3+ to 6 inches, and (3) silt loam @ 6+ inches in depth. As indicated by the varying grey-scale shades at the three levels, the clay content and horizon depth greatly influences the transport of volumetric soil moisture through time. In Appendix Figure (5), the simulation illustrates two point sources of saturated soil at positions (5,5) and (10,9). Again, the "mole-hill" is located at position (5,5) and elevations gradually decline from lower right to upper left. In this simulation the upper crest, pixel (10,9), has distributed its volumetric moisture to adjacent pixels, and moisture is distributed down-slope toward adjacent low areas. Pixel (5,5) remains saturated due to the combined effects of overland flow and high point source precipitation. In the simulation shown in Appendix Figure (6), lower elevation pixels (adjacent to the "mole-hill") are approaching saturation. Locations between the mole-hill and the crest are also becoming saturated, and moisture flow is beginning to divert around the mole-hill toward the drier low lying areas.



Appendix Figure 5. Early simulation matrix for soil moisture in three horizons



Appendix Figure 6. Later simulation matrix indicating complex volumetric soil moisture profile in three horizons

UTILITY OF AN ARTIFICIAL INTELLIGENCE SYSTEM IN FORECASTING OF BOUNDARY-LAYER DYNAMICS

M. D. McCorcle, S. E. Taylor, and J. D. Fast
Iowa State University
Ames, IA 50011

ABSTRACT

Determination of input data requirements for numerical prediction of boundary phenomena and presentation of computational results, in-process subroutine selection and interpretation of model output can be facilitated by techniques of artificial intelligence. A detailed numerical model of the atmospheric boundary layer has been developed and used in research involving low-level transport of momentum, heat, and moisture. The model has shown utility for a number of horizontal scales and has been applied to prediction of regional convergence patterns associated with severe weather, the local transport of agricultural pesticides, and simulation of insect pest transport. This system allows simulation of stability conditions in the three kilometer layer and details energy exchange at the surface. The time-incremental nature of the model predicts the time of maximal transport and minimum transport and may be used for delineating regions of dispersion and of deposition.

1. INTRODUCTION

As meteorological research has advanced, atmospheric models have grown to incorporate many surface physical processes on a wide-range of horizontal scales. These models have been developed to survey microscale processes of turbulence, mesoscale dispersion of pollutants, as well as regional and large-scale circulations leading to severe weather development or transport of biotic agents. Some of these models, such as the one discussed in this report, have shown utility in a variety of situations. Each applied boundary layer problem requires specific initialization procedures, in-process subroutine selection, boundary criteria, and output processing. The time-consuming process of preparing for a particular operational problem may be expedited using artificial intelligence techniques in harmony with a more general boundary-layer forecast system. In addition, verification of model results may be incorporated into an expanded knowledge base which improves the utility of the meteorological model.

The goal of this research has been the development of a versatile forecast system capable of accurate simulation of low-level atmospheric circulations by utilizing fine grid resolution and detailed surface physics. The model has been used in basic research on low-level jets and boundary-layer convergence fields over the

Great Plains and in applied problems involving long-range movement of insect pests to the Corn Belt and the regional transport of agricultural pesticides.

This paper will discuss the status of this research and present future plans for the incorporation of expert system technology. An overview of the boundary-layer forecast system used in this study, found in Section 2, is followed by a discussion of results from three specific applications of the model. Artificial intelligence possibilities are discussed in Section 4.

2. THE MODEL

A three-dimensional numerical forecast model for the planetary boundary layer using a coupled atmosphere-soil system is described in detail in McCorcle (1988). The atmospheric portion of the model is governed by an anelastic, hydrostatic system of equations derived for a terrain-following coordinate system. Temperature, specific humidity, turbulent energy, and the horizontal wind are predicted in the model by using a prognostic equation of the form:

$$\frac{\partial A}{\partial t} + \vec{V} \cdot \nabla A = \frac{\partial}{\partial z}\left[K \frac{\partial A}{\partial z}\right] + F \quad , \qquad (1)$$

where A is the prognostic variable and F is the forcing, which includes the pressure gradient force and buoyancy effects. Typically, this type of equation is discretized by using various finite-differencing approaches. For results reported herein, the advection term, $\vec{V} \cdot \nabla A$, is approximated by using a leapfrog scheme. According to Paegle et al. (1976), finite difference approximations in z prove unstable for many profiles of K because of nonlinear computational instabilities. To overcome this problem, the right-hand side of Eq. (1) is discretized by use of a finite-element technique based upon Galerkin approximations. A detailed discussion of this method is found in Paegle and McLawhorn (1983). The vertical velocity is computed diagnostically and is calculated by integrating the continuity equation by use of centered differencing. The pressure deviation is determined by integrating the hydrostatic equation downward. Physical forcing in the model is due to radiative heating and cooling prescribed at the earth-atmosphere interface in a surface energy budget equation.

To account for the large vertical gradients of many of the prognostic variables at low levels of the atmosphere, a transformed grid system is used to increase resolution in the vertical. The lowest nine levels, below 119 m, use a logarithmically-spaced grid. Above 119 m, 10 additional levels are equally spaced about 205 m apart, up to the top of the model at 2500 m above the surface.

To more precisely predict surface forcings, the model incorporates forecasts of both moisture and heat fluxes within the soil by using a soil-moisture forecast method similar to that

described by Mahrt and Pan (1984) and Pan and Mahrt (1987). The vertical grid consists of 15 soil-temperature computation levels, spaced equally 0.04 m apart, extending from the roughness height at 0.04 m to 0.52 m below the surface. The soil hydrology model uses a two-layer method to update soil-moisture content. The upper layer is 0.08 m deep and the lower layer is 0.44 m. Because temperature forecasts rely on soil-moisture content to calculate soil thermal conductivity and heat capacity, updated soil-moisture values are interpolated to match the soil-temperature forecast levels. The model allows for the addition of parameters to simulate vegetation effects such as transpiration and canopy-water evaporation.

3. MODEL APPLICATIONS

3.1 SIMULATION OF THE GREAT PLAINS LOW-LEVEL JET

The diurnal oscillation of convective events in the Great Plains of North America has been documented by investigators for many years (Blackadar, 1957; Holton, 1967; Bonner, 1968; Wallace, 1975). Their findings have shown that the nocturnal phasing of convection may occur even when the large-scale synoptic situation would warrant convection both day and night. Most theories have attributed these diurnal oscillations to changes in boundary-layer convergence fields resulting from a low-level nocturnal jet. Convergence in the region north of this jet may result in strong upward vertical motion at the top of the boundary layer. Recent studies have been devoted to simulating these oscillatory characteristics in a boundary-layer forecast model.

To examine the model's ability to simulate observed jet phenomena, a springtime case of nocturnal convection is analyzed, and data from this period are used to initialize the forecast model. 26-28 May 1988 was characterized by pronounced nocturnal thunderstorm activity over the central and southern Plains. Rawinsonde observations suggested a low-level wind oscillation during this period. The nocturnal maximum of the winds and the convection could not be explained by the large-scale synoptic situation nor was it forecast well by the National Weather Service forecast models.

A 48-hour forecast was integrated having been intialized with data from 1200 GMT 26 May. The National Weather Service Nested Grid Model forecast of the 850 mb height field was used as the upper-boundary condition to retain realistic ambient conditions. The radar summary at 0535 GMT 27 May depicted in Fig. 1a, shows widespread convection over the northern Great Plains. In Fig. 1b the forecast wind vectors and isotachs at 500 m are shown for the same time period. Here a well-defined jet is forecast over the Central Plains with speeds near 30 m s^{-1}. The low-level convergence associated with this exit region of this wind maximum was a likely trigger for the observed convection. An east-west vertical cross-section of the forecast low-level jet is presented in Fig. 1c.

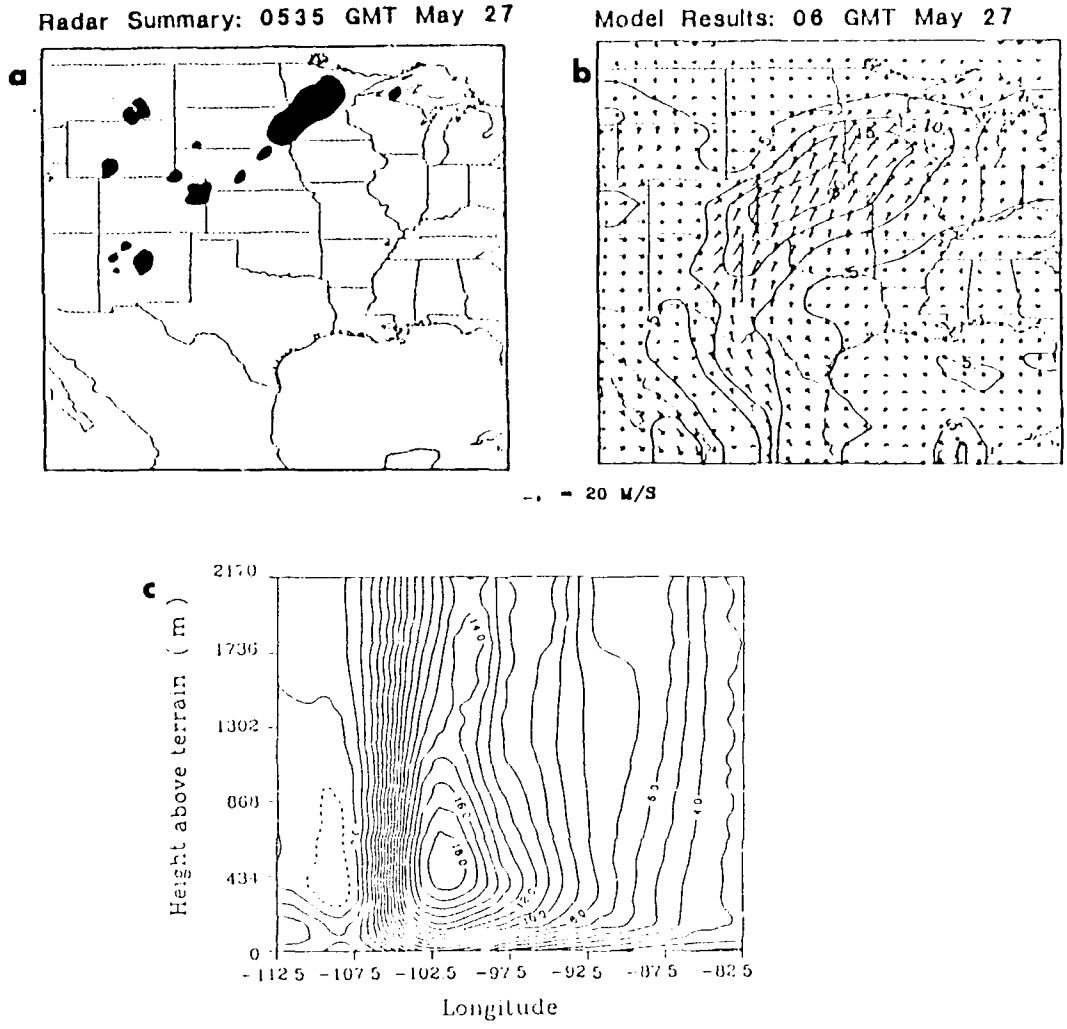


Figure 1. a) Radar summary of precipitation for 0535 GMT 27 May 1988. b) Model forecast wind vectors and isotachs in ms^{-1} valid 0600 GMT 27 May. c) East-west vertical cross-section of the v component (north-south) wind speed at the Kansas-Oklahoma border (37°N latitude) valid 0600 GMT 27 May. Solid contours denote southerly sinds and dashed denote northerly winds,

3.2 REGIONAL TRANSPORT OF INSECT PESTS

Some insect pests of corn are introduced each spring to the Midwest by the northward migration of populations that overwinter far to south. Research by Kaster and Showers (1982) and Domino et al. (1983) has shown that nocturnal, long-range movement of noctuids, such as the black cutworm, *Agrotis ipsilon* (Hufnagel), is strongly correlated to particular low-level wind conditions that are

common during the spring over the central United States. This research has attempted to formulate a forecast method to accurately predict the introduction of wind-transported pests. Forecast information of insect introduction may then be used for insecticide planning and other pest management decisions.

An advection-diffusion forecast routine has been developed and added to the numerical forecast system described in Section 2. The method directly predicts potential pest concentrations throughout the domain. To simulate nocturnal flight characteristics of the black cutworm moth, three-dimensional dispersion in the model ceases after sunrise and concentrations are summed vertically and held constant until after sunset when dispersion resumes. Factors such as swarming and flight speed are introduced directly into the concentration forecast equation by varying the turbulent diffusion coefficient and the advecting windspeed, respectively. Initial concentrations are determined from winter soil temperature extremes, accumulated growing degree days, trapping observations, and continuity from earlier forecasts.

Black cutworm moth concentration estimates from the dispersion forecast for 22-24 March and 7-9 May 1988 are compared with trapping observations in Figs. 2 and 3, respectively. (Note that Missouri data is not yet available for these periods.) These results agree with trapping observations, confirming that the introduction of moths to Iowa and Illinois may have occurred as soon as two nights after the onset of the southerly winds. For both cases the highest black cutworm density was found near the low-level

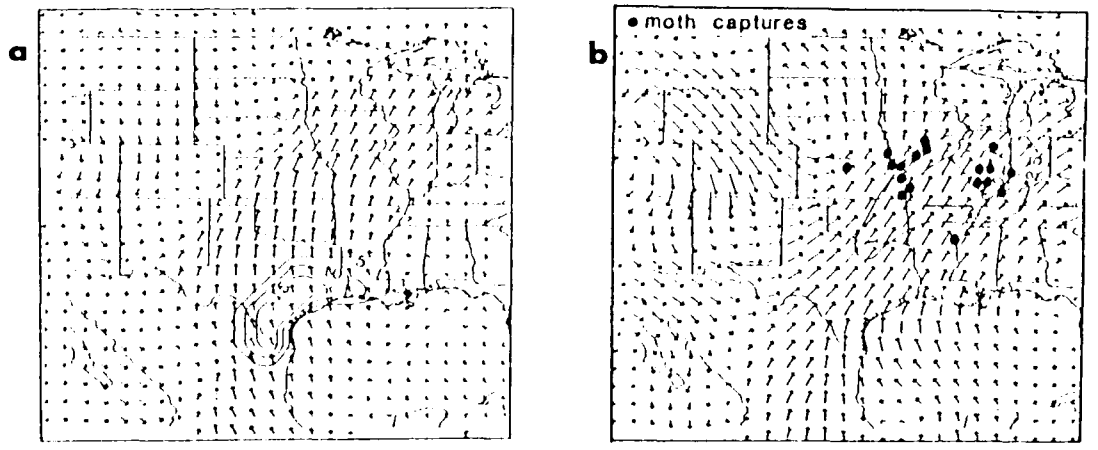


Figure 2. Model forecast 500-m wind vectors and computed concentration index[1] valid a) 1800 CST, 22 March and b) 0600 CST, 24 March (solid circles denote trapped black cutworm moths, Missouri not reporting).

[1]The concentration index is a dimensionless quantity based only on the initial field.

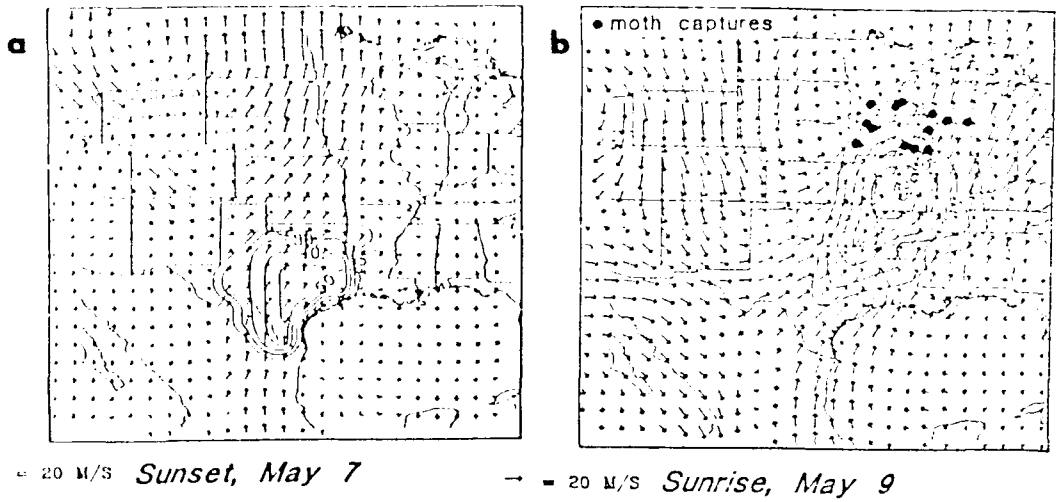


Figure 3. Model forecast 500-m wind vectors and concentration index valid a) 1800 CST, 7 May and b) 0600 CST, 9 May (solid circle denote trapped black cutworm moths).

wind maximum, or jet, at 500 meters above the ground. These results agree with those of Wolf et al. (1986) who reported that radar-observed flight tracks were near the 500-meter level.

3.3 DISPERSION OF AGRICULTURAL PESTICIDES

Selection of initial conditions and model elements permits another dispersion simulation method which has been applied to the mesoscale movement of volatilized agricultural pesticides. The impact of these chemicals on water and air quality in areas in and

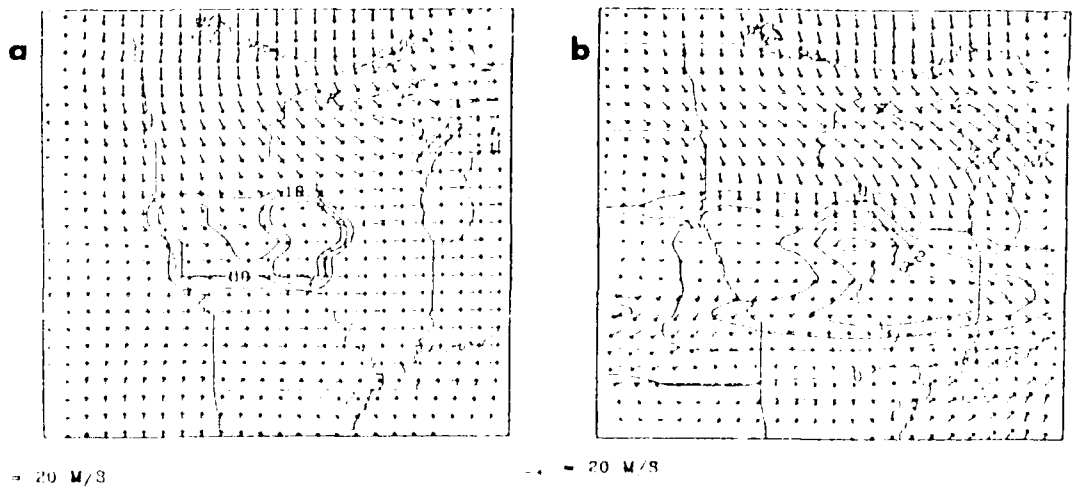


Figure 4. a) Initial pesticide concentration index and wind vectors at the surface. b) Twenty-four-hour model forecast of 500-m concentration index and wind vectors.

adjacent to the Corn Belt is an increasing health concern. Figure 4 shows wind and concentration index results of a twenty-four hour simulation of pesticide movement out of Iowa. A finer grid resolution was chosen for this problem. Initialized concentrations were based on typical springtime pesticide applications and volatilization rates for the given meteorological conditions. The synoptic situation was based on a 1988 springtime case and data from the Nested Grid Model was once again used for the upper boundary condition for pressure.

4. INCORPORATION OF ARTIFICIAL INTELLIGENCE TECHNIQUES

4.1 MODEL INITIALIZATION

The numerical model can be initialized for a range of grid resolutions. A coarse grid, 150 km, is chosen for simulation of synoptic-scale, low level, circulation. Time increments of 500-1000 s will satisfy numerical stability criteria. Rawinsonde observations are normally adequate for initial temperature, moisture, wind and pressure field inputs. When the computational model is used for regional, sub-syroptic, descriptions, a finer grid, decreased time step, and incorporation of surface weather observations becomes important. Movement of biotics is usually analyzed at the sub-synoptic scale Drift of chemical agents such as broadcast pesticides requires grid scales less than 75 km and greater terrain analytic detail. The model has been applied to a scale as fine as 20 km. Vertical resolution both in the atmosphere and in the soil may be adjusted as desirable.

A knowledge-based system can be implemented to determine the optimal grid size for a specified application. The system will additionally recommend or specify all other initial configuration parameters and data requirements including appropriate time steps and terrain field data. If the user is unable to provide all recommended inputs, the expert system will give an evaluation of impact associated with non-optimal configurations.

4.2 RUN-TIME DECISIONS

The configuration of the process used in execution has considerable influence on computational time and on the precision and accuracy of numerical results. A number of subroutines dealing with soil moisture, surface heating, localized precipitation, time-dependent dispersion, and computation by finite-differencing or by finite element techniques may be selected by logical optimization.

4.3 OUTPUT PRESENTATION

Graphic and tabular model output should be optimized to present the precision appropriate to the intended use of the simulation. Output may be directed to a printer, file, or graphic plotting device and depict wind, pressure, temperature, humidity, or dispersoid concentrations at appropriate scales and accuracy. The

selection of output parameters is rule-governed. Accordingly, output selection may be achieved under system control.

4.4 OUTPUT INTERPRETATION

When model generated output has been verified by observation, some variability between theoretical and real-world occurrence is expected. A knowledge-based matrix will enable the system user to comment on differences from reality and to specify parameters that the user suspects may be influencing the system. Such user-provided observations will provide a probability analysis and bias corrected output which improves model reliability as implementation expands.

5. SUMMARY

Assessment of specifications that are available for applying a boundary-layer forecast system to a range of perceived functions, indicates that expert system technology can provide substantially increased system utility. A three-dimensional, coupled earth-atmosphere model to forecast boundary layer phenomena has been applied to describe a number of meteorological and biological events. The configuration of model elements and input/output parameters required to exploit the capability of the numerical method normally requires the efforts of a highly-trained specialist. Artificial intelligence programming is conducive to the creation of a meteorological system that is adapted to the potential applications by non-expert, but competent, personnel. Such a system can provide both numerical and non-numerical output as required by the user.

ACKNOWLEDGEMENTS

This research has been funded by USDA-CSRS Contract No. 87-CRSR-2-3003 and the Iowa State University Home Economics and Agicultural Experiment Station Project No. 2804.

REFERENCES

Blackadar, A.K., 1957: Boundary layer wind maxima and their significance for the growth of nocturnal inversions. *Bull. Amer. Meteor Soc.*, **38**, 283-290.

Bonner, W.D., 1968: Climatology of the low level jet. *Mon. Wea. Rev.*, **96**, 833-850.

Domino, R. P., W. B. Showers, S. E. Taylor and R. H. Shaw. 1983. Spring weather pattern associated with suspected black cutworm moth (Lepidoptera: Noctuidae) introduction to Iowa. *Environ. Entomol.*, **12**, 1863-1872.

Holton, J.R., 1967: The diurnal boundary layer wind oscillation above sloping terrain. *Tellus*, **19**, 119-205.

Kaster, L. V. and W. B. Showers. 1982. Evidence of spring immigration and autumn reproductive diapause of the adult black cutworm in Iowa. *Environ. Entomol.*, **11**, 306-312.

Mahrt, L. and H. Pan, 1984: A two-layer model of soil hydrology. *Boundary-Layer Meteorol.*, **29**, 1-20.

McCorcle, M. D. 1988. Simulation of surface moisture effects on the Great Plains low-level jet. *Mon. Wea. Rev.*, **116**, 1705-1720.

Paegle, J. and D. W. McLawhorn. 1983. Numerical modelling of diurnal convergence oscillations above sloping terrain. *Mon. Wea. Rev.*, **111**, 67-85.

Paegle, J., W.G. Zdundowski and R.M. Welch, 1976: Implicit differencing of predictive equations of the boundary layer. *Mon. Wea. Rev.*, **104**, 1321-1324.

Pan, H.-L., and L. Mahrt, 1987: Interaction between soil hydrology and boundary-layer development. *Boundary-Layer Meteorol.* **38**, 185-202.

Wallace, J.M., 1975: Diurnal variations in precipitation and thunderstorm frequency over the conterminous United States. *Mon. Wea. Rev.*, **103**, 406-419.

Wolf, W. W., J. K. Westbrook, and A. N. Sparks. 1986. Relationship between radar entomological measurements and atmospheric structure in south Texas during March and April 1982. pp. 84-97. In: A. N. Sparks (ed.), Long-range migration of moths of agronomic importance to the United States and Canada: Specific examples of occurrence and synoptic weather patterns conducive to migration. U.S. Dept. of Agriculture, Agricultural Research Service ARS-43, 104 pp.

RESEARCH IN TERRAIN KNOWLEDGE REPRESENTATION FOR IMAGE INTERPRETATION AND TERRAIN ANALYSIS

Olin Mintzer
Center for Autonomous Technologies
Research Institute
U. S. Army Engineer Topographic Laboratories
Fort Belvoir, Virginia 22060-5546

ABSTRACT

This research paper emphasizes use of landforms as a fundamental component for the representation and organization of terrain knowledge. Explicit characterization of observable terrestrial patterns provides an intelligent basis for the identification of landforms from stereoscopic aerial imagery. Knowledge of properties associated with individual landforms provides a foundation from which to infer terrain aspects of military relevance. On-going research focuses on the development of an expert system for landform identification. Here, a computer program serves as an on-line computer consultant to a terrain analyst interpreting stereoscopic aerial photography. An expert system shell is used as a tool to encode domain specific knowledge relating photo-identifiable features to specific landforms. A second aspect of the research, not yet implemented as an expert system, deals with the potential to use landform information to support battlefield decisionmaking. Consider the following example set in an arid environment with such landforms as playa, alluvial fan and hills where a knowledgeable terrain analyst can make the significant predictions based on general properties associated with landforms. On a playa, tanks will bog down under wet conditions, helicopter landing zone operations will be impaired by dust and no aggregates will be available for construction. On an alluvial fan, gullies can conceal tanks as well as deter tank movements when the gullies are perpendicular to the avenue of approach. Adjacent pairs of hills will create choke points. Thus, knowledge of the properties of individual landforms provides a useful approach to address general questions of military relevance. These observations suggest that landform characterization and classification provide a potentially rich, generalized frame of reference for high-level representation and organization of terrain information. This approach may be of particular value when detailed Defense Mapping Agency terrain analysis products such as the Tactical Terrain Analysis Data Base (TTADB) or digital Tactical Terrain Data (TTD) are not available.

1. INTRODUCTION

The focus of this paper is the terrain analyst's characterization and classification of landforms, an important aspect of the art and science of terrain analysis. The use of landforms is emphasized as a critical component for the representation and organization of terrain knowledge. Explicit characterization of observable terrestrial patterns provides an intelligent basis for the identification of landforms from imagery; moreover, knowledge of the identity of a landform provides the basis to infer many of its significant military aspects. This research is based on the development of an experimental expert system to assist the terrain analyst in identifying a given landform from stereoscopic aerial photography. The approach is outlined in figure 1. The resulting interactive software provides a useful vehicle to encode terrain information and to explore fundamental issues of terrain knowledge representation and exploitation.

The criteria used to identify landforms is adapted from an engineering approach developed for manual photo interpretation in which landforms are characterized by pattern elements. In Section 2, the pattern elements are defined and some of their key properties are described.

Construction and operation of the experimental expert system is presented in Section 3. This section deals with how descriptors based on landform pattern elements are incorporated as a knowledge base in a domain-independent expert system shell. Then, use of the expert system is demonstrated with two examples in which landforms are identified based on user-entered values for climatic and photo-observable terrestrial pattern elements.

In Section 4, potential use of generalized properties associated with landforms is discussed in the context of several examples. Finally, this approach to terrain knowledge representation for image interpretation and terrain analysis is summarized in Section 5.

2. DOMAIN KNOWLEDGE: LANDFORM CHARACTERIZATION

Landforms are the expression of physical relief of the land surface as developed by erosional or depositional processes under given climatic and geologic conditions (Frost et al., 1953). The landform is identifiable on imagery as a singular feature by stereoscopic observation and often is uniquely characterized by a convergence of evidence (Pinker and Tori, 1984). In an engineering approach to photo interpretation, Minzer and Messmore (1984) identified a set of key pattern elements for terrain classification: climate,

formtype, drainage, density, erosion, special, tone, land use and vegetation. Taken together, descriptions based on these pattern elements provide valuable indicators to the identity of an unknown landform.

Regions of the earth are classified in terms of climate *(climate)* based upon knowledge of specific temperature, precipitation and environmental characteristics. It is desirable to limit the set of possible landform candidates t those which are known to exist in the specific climatic region.

Each landform has a characteristic geometry or form (*formtype*). The surface materials of landforms have specific modes of deposition or occurrence. In general, landforms are quite homogeneous in texture and composition being derived from the same unconsolidated or consolidated materials; often, the surface materials of the landforms are the parent materials.

The surface drainage (*drainage*) is useful in determining the general, shallow sub-surface texture, composition and distribution of the surface materials, and the topographic relief of an area. The drainage represents the impression rainfall has had on the runoff pattern of the eroded surface. Where overburden is thin, drainage patterns provide general information about the structural attitude of the consolidated materials.

Density (*density*) refers to how closely or openly spaced the drainageways are from each other. In general, the description of the spacing ranges from fine to coarse. Usually, a fine-spaced drainageway system signifies that a lot of erosion has occurred -- silty or clayey soils are indicated.

Erosional characteristics (*erosion*) are important indicators of the surface texture, composition and distribution of the surface materials. Gullies with V-shaped profiles usually indicate medium-to-coarse textured soils, while U-shaped or saucer-shaped gully profiles indicate silty or clayey textures. Erosional features assist in identifying the composition, texture and layering of the parent materials. An example is a dissected, thick shale sequence that usually exhibits smooth slopes and subdued, rounded topography of moderate relief.

Special features (*special*) include features which may be described under some other pattern element; often, they assist in identifying a landform. Orchards, for example, provide an indication of granular surface materials of good internal drainage. Field tile and ditches indicate low relief and/or poor surface drainage. Poor drainage, along

with the characteristic tonal patterns, indicate soft subsoils.

Photo tones (*tone*) are indicators of texture, distribution and composition of the surface materials, and at times indicate the moisture condition. Tones often indicate the relief and drainage characteristics, and are essential in the identification of the landforms which possess characteristic tonal patterns. In general, light photo tones indicate granular, well-drained soils, while dark tones indicate fine-textured, poorly-drained soils. In agricultural areas, gray tone patterns that are large and rectangular in shape, indicate that the area is relatively flat. On the other hand, irregular, narrow, banded tonal patterns due to contour farming indicate high or moderate relief.

Land use (*land*) is an important indicator of relief in an area. Flat areas are generally under cultivation, while upland terrain is identified by large wooded areas, limited farming and a predominance of pasture land.

Vegetation (*veg*) is an indicator of the texture and drainage characteristics of the surface materials. Orchards are indicative of sandy, well-drained soils, while willows and swamp vegetation are indicative of poorly-drained, fine-textured surface materials.

Based on pattern element analysis and collateral data, the engineering photo interpreter systematically delineates and classifies terrain features. Identifications of landform-associated surface materials are considered reliable, but many subsurface and stratigraphic interpretations require more detailed analyses.

To formalize these concepts in an expert system, landform fact sheets were assembled describing photo-observable characteristics for a set of major landforms from a data set of representative stereo aerial photography. Analysis of several approaches to landform characterization led to selection of the terminology described above. The descriptive categories continue to be studied and revised for precision and conciseness.

3. AN EXPERT SYSTEM FOR LANDFORM IDENTIFICATION

3.1 DESCRIPTION OF AN EXPERIMENTAL EXPERT SYSTEM

The expert system, as discussed in this paper, is an interactive computer program that uses reasoning techniques to solve problems operating on a problem domain knowledge base with case specific input provided by a user (Waterman and Hayes-Roth, 1982). Expert knowledge including problem domain descriptions (facts) and heuristics (rules of thumb)

is assembled in a knowledge base. The domain-independent reasoning mechanism that operates on knowledge base and case attributes is known as a inference engine (Van Horn, 1986). In addition to suitable knowledge representation, the inference engine requires a strategy to address individual problems and a mechanism to reason with uncertain data.

The task addressed in this expert system research is landform identification. It is designed to assist a terrain analyst in the interpretation of stereoscopic aerial photography, and provides an excellent research vehicle to explore and test terrain knowledge representation. Our approach embodies problem domain knowledge adapted from the pattern element analysis approach described in Section 2. The goal is to identify individual landforms observed in stereo aerial photography based on evidence elicited from the terrain analyst.

3.2 KNOWLEDGE ENGINEERING SYSTEM (KES)

Knowledge Engineering System (KES) software, a domain-independent expert system shell, was installed in the Center for Artificial Intelligence, Research Institute, Engineer Topographic Laboratories on a VAX 11/780 in 1985. Originally developed in LISP, KES Version 2.4 has been implemented in the C programming language and is available for a range of micro, mini and mainframe computers. KES supports three approaches to decisionmaking: production rules (PS); hypothesis-and-test (HT); and Bayesian statistical pattern classification (Software A & E, 1987a,b).

KES modules allow a problem domain specialist to create, test and refine a domain specific knowledge base which incorporates class attributes and values. Rules are instantiated for each member of a class. Facts are declared as having single or multiple values. The rules are written in logical form (IF-THEN-ELSE) providing a basis for operating on individual case data. KES supports an approach to the definition of classes and class attributes that is easy to use and understand. A special class section in the knowledge base describes the attributes in each class.

3.3 BUILDING A KNOWLEDGE BASE FOR LANDFORM IDENTIFICATION

To construct a knowledge base, landform class descriptions similar to those used in manual terrain analysis processes are specified as attributes and characterized by possible values. A priori physiographic knowledge (Fenneman, 1931) is invoked to establish a set of landforms that can be expected in a given geographic region. To date, this research has studied 34 different landforms derived from unconsolidated materials of glacial, eolian and fluvial origin. The current set of landform attributes and values encoded in the knowledge base is shown in Figure 6.

Consistent with Section 2, the set of nine attribute classes are maintained ranging from climate (*climate*) to vegetation (*veg*). *Drainage*, for example, is characterized by a set of 17 possible values ranging from *pinnate* to *artificial*. Each value is relevant to at least one, and generally, many more landforms.

Instantiations for two specific landforms, alluvial fan and playa, have been extracted from the knowledge base and presented in figure 3. Note that in these cases, a single value has been associated with each of the nine landform descriptions. In the case of the *alluvial fan* landform, *erosion* is described exclusively as *v_shaped* while the value for *erosion* on a *playa* is characterized as *none*.

3.4 USING AN EXPERT SYSTEM FOR LANDFORM IDENTIFICATION

To use the KES expert system, the user works from an alphanumeric terminal placed adjacent to an airphoto image display (a film-based light table or digital image processor). Two examples of aerial photos from a desert environment in the southwestern United States are shown in figures 4 and 5. The computer program addresses the user with a series of queries to eliciting case specific values for each relevant attribute. The inference engine operates on the information furnished by the user and the knowledge base to identify the landform. If all attribute values match a landform description in the knowledge base, KES returns the name of the identified landform and the annotation *<a>* which represents a certainty factor of "always". If no single landform description is consistent with all the user specified observations, the program returns a ranked list of candidate landforms using the notation *<h>*, *<m>* and *<l>* to indicate a "high", "moderate" or "low" belief in the potential identification.

Portions of a typical interactive session are recorded in figure 6. The user initiates the session and is asked to enter a value for *climate* from a menu of 6 possible conditions. In this case, option 6 indicating *arid_semi_arid* is specified, then the user is prompted for a value for *formtype*. Note that for this attribute, and many others, multiple values may be entered. Here, the user has examined the imagery and then entered option 8 indicating *fan shaped plain*. The dialog continues until values are instantiated for all nine attributes. In this example, each of the entries corresponds to attribute values in the knowledge base for alluvial fan (figure 3) and the inference engine has returned the identification *alluvial fan* with a certainty factor of *<a>*.

A portion of an interactive session associated with the aerial photo in figure 5 is shown in figure 7. Here, the

user exercises the capability to enter multiple values for the *special* feature attribute. Options 19, 24 and 25 are selected which represent the conditions *bounded by uplands, alkali deposits* and *oval shaped depressions*. This set of entries trigger another feature of KES. The set of values specified by these entries and the previously entered values are not consistent with the description of any one landform represented in the knowledge base. The user is given an opportunity to change the entries or continue. In this example, the user elected to continue. After values for all attributes have been specified, the inference engine returned a list headed by *playa <h>* indicating a high degree of certainty, followed by 33 less probable landforms, each with a low certainty factor (*<l>*). The landform is, in fact, a playa.

4. PREDICTION BASED ON LANDFORM PROPERTIES

General properties that are significant for military planning can be assigned to each category of landform. When a terrain analyst identifies a particular landform, he can invoke knowledge associated with specific landforms in light of current environmental conditions to support battlefield decisionmaking. Consider the following examples to illustrate the potential use of such a priori knowledge.

In one of the previous examples, the expert system led to the identification of an alluvial fan in an arid environment. Based on general properties associated with this condition, the terrain analyst can make several assertions. This landform provides aggregate materials for construction of surfaces and subsurfaces of roads and runways. Cross country mobility will be delayed by the distribution of deep gullies intersecting the avenues of approach. Tanks will have to go around the alluvial fan, thus taking a longer route of travel. Deep gullies provide cover for tanks and reconnaissance patrols.

The second example presented earlier, identified a playa in an arid environment. In this case, the terrain analyst considers the potential for soft soils, dust and construction aggregates -- he can predict that on a playa, tanks will bog down under wet conditions, helicopter landing zone operations will be impaired by dust and no aggregate material will be available for construction.

This approach is well-suited to other environmental situations. Suppose the terrain has been identified as an igneous region of volcanics -- cinder cones in an arid environment. Along with the cinder cones, there will be boulder fields. These features will present obstacles to movement as well as cover for troop movement on the ground. Rock outcrops occur on steep slopes presenting barriers to tank movement up or down slopes.

In this manner, terrain characterization based on landform can be used to establish a generalized knowledge base from which high-level inferences may be drawn. This approach may be of particular value when detailed Defense Mapping Agency terrain analysis products such as the Tactical Terrain Analysis Data Base (TTADB) or digital Tactical Terrain Data (TTD) are not available.

5. SUMMARY

This paper presents a research effort concerned with building an expert system to identify landforms for the terrain analyst. It has shown how the descriptors from observing imagery are used in developing a knowledge base using an expert system. The landform descriptors were then converted into the knowledge base using a hierarchical format. The Knowledge Engineering System (KES) was run during which inferences were made as to the identities of the observed unknown landforms. Given the knowledge of the landforms, the terrain analyst provides the commander with terrain knowledge that enables him to make decisions based on the Intelligence Preparation of the Battlefield (IPB).

ACKNOWLEDGMENTS

The on-going research described in this paper is conducted at the Research Institute, U. S. Army Engineer Topographic Laboratories (USAETL). Portions of this paper were prepared at the Center of Automated Image Analysis under the supervision of Dr. Frederick Rohde and at the Center for Autonomous Technologies under the supervision of Mr. George E. Lukes. Mr. Lawrence A. Gambino served as Director, Research Institute during this period. COL David F. Maune is the Commander and Director, USAETL, and Mr. Walter E. Boge is the Technical Director.

The author appreciates the assistance provided by Mr. George E. Lukes, Mrs. Mary Puccia and Mrs. Marion Mintzer in the preparation of this paper.

REFERENCES

Fenneman, N. M., 1931: Physiography of Western United States, McGraw-Hill Company, New York, NY.

Frost, R. E., J. G. Johnstone, O. W. Mintzer, M. Parvis, P. Montano, R. D. Miles and J. P. Shepard, 1953: A Manual of Airphoto Interpretation of Soils and Rocks for Engineering

Purposes, School of Civil Engineering and Engineering Mechanics, Purdue University, West Lafayette, IN.

Mintzer, O. W. and J. A. Messmore, 1984: "Terrain Analysis Procedural Guide for Surface Configuration," Report No. ETL 0352, U. S. Army Engineer Topographic Laboratories, Ft. Belvoir, VA.

Rinker, J. N. and P. A. Corl, 1984: "Airphoto Analysis, Photointerpretation Logic, and Feature Extraction," Report No. ETL 0329, U. S. Army Engineer Topographic Laboratories, Ft. Belvoir, VA.

Software Architecture & Engineering, 1987: Knowledge Base Author's Manual, Arlington, VA.

Software Architecture & Engineering, 1987: Knowledge Base Reference Manual, Arlington, VA.

Van Horn, Mike, 1986: Understanding Expert Systems, The Waite Group, Bantam Books, New York, NY.

Waterman, D. A., Hayes-Roth, Frederick, 1982: "An Investigation of Tools for Building Expert Systems," prepared for the National Science Foundation, The Rand Corporation, Santa Monica, CA.

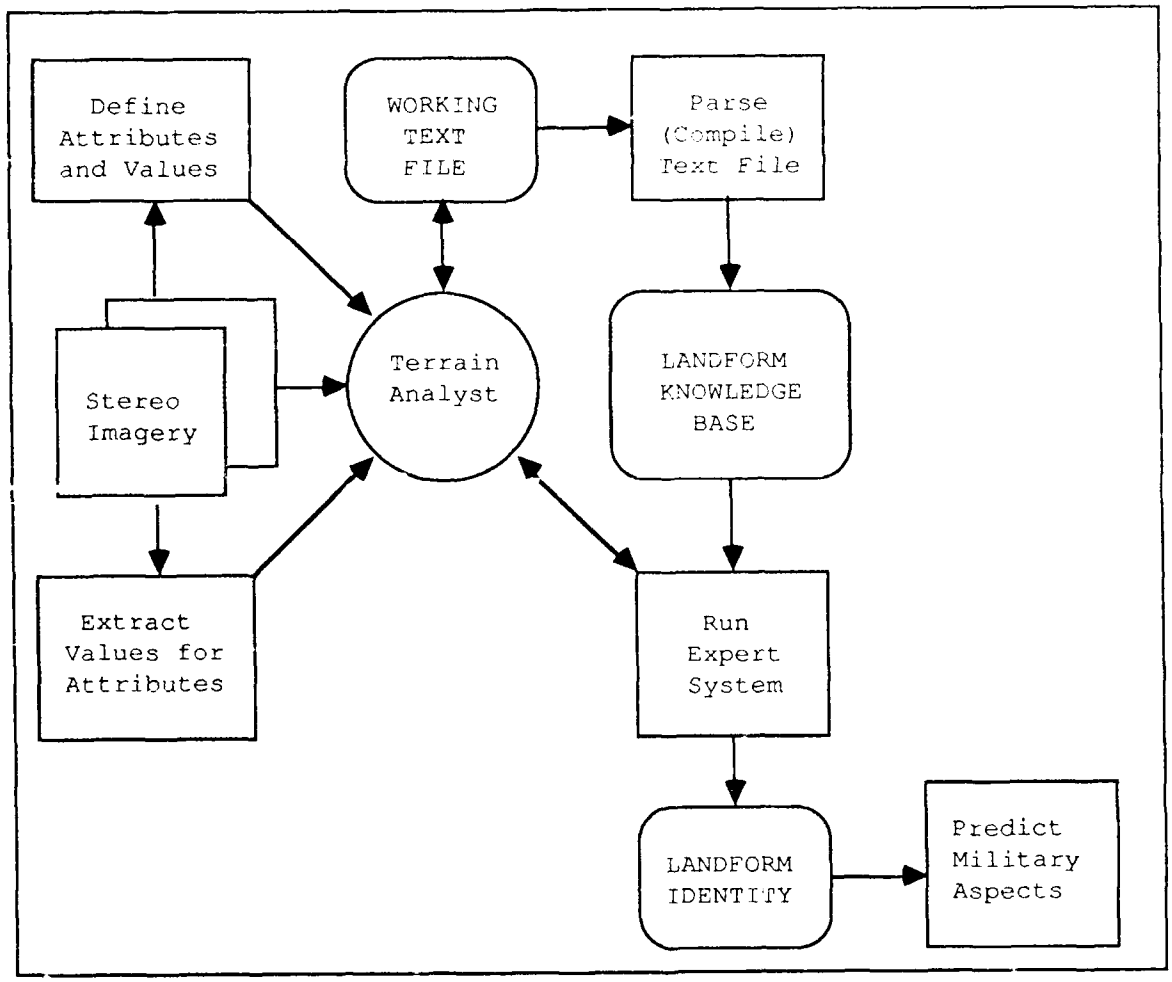


FIGURE 1. Approach to landform identification and prediction.

```
attributes:
   climate: sgl
      (semi_humid, semi_arid, humid, all, arid, arid_semi_arid).
   formtype: mlt
      (plain, ridge, hill, hummock, sloping plain, ridges, channel,
      fan_shaped plain, arcuate plain, dissected cuestas,
      ridge_plain, broad plains, knob_plain).
   drainage: mlt
      (pinnate, dendritic, absent, internal, parallel braided,
      parallel, meanders, radial_braided, anastomotic,
      dendritic distributary, crisscrossing_branching,
      deranged_dendritic, artificial).
   density: sgl
      (none, medium, coarse, fine, coarse to medium).
   erosion: mlt
      (u_shaped, sag_swale, box shape, none, v_shaped,
 saucer_shaped, u_shaped_saucer_shaped, v_shaped_u_shaped).
   special: mlt
      (conical mound on plain, cat steps, crescent shapes,
      low relief, blowouts, cliffs adjacent to water, dissected,
      contour plowing, wave pattern, star shapes,
      dune remnants, equal sideslopes, snakelike,
      fluvial marks, cigar_shaped, deranged_medium dendritic,
      meanders_abandoned channels_natural levees,
      delta shape, bounded by uplands, fan_shaped, coalescing
      fans, buffalo wallows, deposition from surrounding uplands,
      alkali deposits, oval shaped depressions, cuestas, parallel
      ridges, meandering drainage, poor drainage, well_drained,
      crisscrossing_braided, broad field pattern).
   tone: mlt
      (light to dark crop, light crop, dark, light ridges,
      dark swales, medium to dark, dark base_light top,
      light with dark streaks, uniform_dull, scrabbled,
      dull plains_dark depressions, light, dark depressions,
      light_dark, complex, light_medium, mixed, dull).
   land: mlt
      (not cultivated, cultivated, natural cover,
      pasture, rangeland, forested, none, borrow pits,
      natural_cultivated, natural_irrigated cultivation).
   veg: mlt
      (grass, scattered trees, dense woods, barren, pasture,
      cleared for cropland, grassland, natural_irrigated
      cultivation, marsh grass, none, sparse natural,
      swamps, forest, scrub growth, marshes).
```

FIGURE 2. Attribute section of knowledge base defining valid values.

```
landform: sgl (
  alluvial fan
    [description:
     climate = arid_semi_arid;
     formtype = fan_shaped plain;
     drainage = dendritic
                distributary;
     density = medium;
     erosion = v_shaped;
     special = coalescing fans;
     tone = light;
     land = natural cover;
     veg = sparse natural;],
```

```
landform:  sgl (
  playa
    [description:
     climate = arid;
     formtype = plain;
     drainage = absent;
     density = none;
     erosion = none;
     special = alkali deposits;
     tone = scrabbled;
     land = natural_irrigated
            cultivation;
     veg = none;],
```

FIGURE 3. Portion of knowledge base providing descriptions for alluvial fan and playa landforms.

FIGURE 4. Aerial photograph of alluvial fan landform.

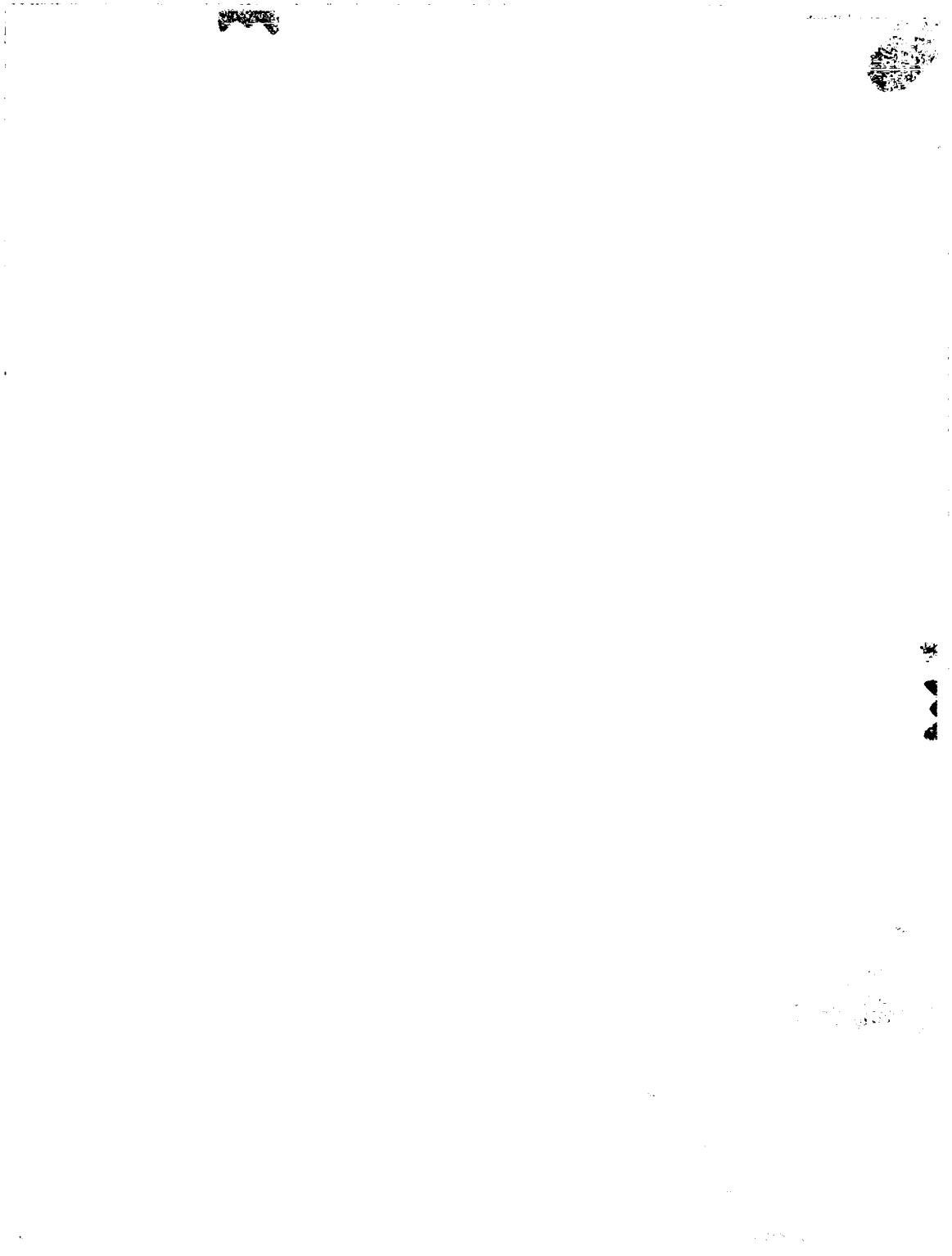


FIGURE 5. Aerial photograph of playa landform.

```
Ready for command: n

climate
        1. semi_humid
        2. semi_arid
        3. humid
        4. all
        5. arid
        6. arid_semi_arid
=? 6

formtype
        1. plain
        2. ridge
        3. hill
        4. hummock
        5. sloping plain
        6. ridges
        7. channel
        8. fan_shaped plain
        9. arcuate plain
       10. dissected cuestas
       11. ridge_plain
       12. broad plains
       13. knob_plain
(multiple answers allowed)
=? 8

drainage
        1. pinnate
        2. dendritic
        3. absent
        4. internal
        5. parallel_braided
        6. parallel
        7. meanders
        8. radial_braided
        9. anastomotic
       10. dendritic distributary
       11. crisscrossing_branching
       12. deranged_dendritic
       13. artificial
(multiple answers allowed)
=? 10

   << density dialog deleted >>

erosion
        1. u_shaped
        2. sag_swale
        3. box shape
        4. none
        5. v_shaped
        6. saucer_shaped
        7. u_shaped_saucer_shaped
        8. v_shaped_u_shaped
(multiple answers allowed)
=? 5

<< special & tone dialog deleted>>

land
        1. not cultivated
        2. cultivated
        3. natural cover
        4. pasture
        5. rangeland
        6. forested
        7. none
        8. borrow pits
        9. natural_cultivated
       10. natural_irrigated
           cultivation
(multiple answers allowed)
=? 3

veg
        1. grass
        2. scattered trees
        3. dense woods
        4. barren
        5. pasture
        6. cleared for cropland
        7. grassland
        8. natural_irrigated
           cultivation
        9. marsh grass
       10. none
       11. natural cover
       12. sparse natural
       13. swamps
       14. forest
       15. scrub growth
       16. marshes
(multiple answers allowed)
=? 12

alluvial fan <a>
```

FIGURE 6. Menu excerpts for alluvial fan identification.

```
special
        1. conical mound on plain
        2. cat steps
        3. crescent shapes
        4. low relief
        5. blowouts

  <<< dialog deleted >>>

       18. delta shape
       19. bounded by uplands
       20. fan_shaped
       21. coalescing fans
       22. buffalo wallows
       23. deposition from surrounding uplands
       24. alkali deposits
       25. oval shaped depressions
       26. cuestas
       27. parallel ridges
       28. meandering drainage
       29. poor drainage
       30. well_drained
       31. crisscrossing_braided
       32. broad field pattern

(multiple answers allowed)
=? 19&24&25

A single value of landform
cannot explain all of the features of this case.

Should processing continue using those values that
are not categorically rejected

=? (y/n) y

  <<< dialog deleted >>>

playa  <h>
stream channel  <1>
esker  <1>
drumlin  <1>
outwash plain  <1>
moraine  <1>
lake bed  <1>
kame terrace  <1>
loessial_plain1  <1>

 <<< 25 lower ranked landforms deleted >>>
```

FIGURE 7. Menu excerpts for playa identification.

IMPROVED EXPERT SYSTEM PERFORMANCE THROUGH KNOWLEDGE SHAPING

Joseph A. Vrba and Juan A. Herrera
Perceptics Corporation
Artificial Intelligence Applications Division
Knoxville, TN 37933-0991 U.S.A.

A primary motivation for developing rule-based expert systems was the desire to separate the specification of the knowledge of a problem from the implementation of its solution. The logic of the problem is encoded into an expert knowledge base. From this abstract form, a procedural version (the executable expert system) which applies the knowledge can be inferred by a computational machine called an inference engine. In software development terms the rules can be viewed as playing the role of a system specification. The inference engine performs the amazing feat of producing a procedural behavior directly out of the specification.

Of course this is too good to be true. There is an important ingredient missing, *control*. In this context, control means the appropriate ordering of computational *events, e.g. causing the rules to fire in the correct* order. Although an inappropriate ordering will often have a dramatic effect, it has proved to be very difficult to determine what it is that *must* be specified over and above the logic to secure control. In a battlefield environment, determinism of an autonomous decision-making system is essential. Without careful imposition of control, real-time performance will surely be compromised.

Recent developments in discrete decision theory [1] give a much clearer idea of what a control specification might look like. They indicate that it is possible (and even *practical*) to separate the control specification from the logical specification. The conventional view of knowledge acquisition concerns primarily the acquisition of the logical specification of the problem, the knowledge base. The control specification is almost always an afterthought. Only after most of the logical knowledge has been acquired does the problem of how it should be controlled become critical. There may be some obvious *features, such as relative expense of making decisions* and the relative likelihood of certain events which the designer might impart to the system. However, a control specification, developed in this fashion, will almost certainly be incomplete and will allow many alternate procedural forms.

With existing expert system shells there are some rather limited methods for incorporating control. In forward chaining, control is relegated to the *conflict resolution* strategy which determines which of the active rules is actually to be fired next. For example, one common conflict resolution strategy is to always prefer more specific rules (i.e. those with more antecedents) over less specific rules. This allows the system builder to write branching behavior more succinctly: the default in the conditional branching becomes the most general case with the more specific cases caught by more specific rules. In backward chaining a worse situation exists: the control is hidden in rule and antecedent ordering. The meaning of a program can be affected by the order in which the rules are expressed! Developers of expert system shells quickly noted control weaknesses and added meta-rules in an attempt to keep the control information separate from the logical specification. It is not clear that this approach really worked.

There is another, much more commonly used mechanism for imposing a control structure on an expert system. A developer will often write the rules to explicitly include an antecedent which determines the control context in which the rule is to be applied (the added antecedent has no *logical* effect). The problem is that such modifications cause the knowledge base to contain much more than abstract logical knowledge. In fact, when one inserts control information into the rules in this fashion, it can no longer be guaranteed that they can be manipulated according to the rules of logic. There is always a danger that a simple logical manipulation will actually completely change the meaning of the procedure generated under the chosen inference mechanism.

It may not be immediately obvious why a logical specification does not provide all the information required to give an acceptable procedural form. In order to illustrate this, consider the following problem :

> Prior to 1965, the Soviet Union deployed three types of main battle tanks; the T-62, the T-55 and the PT-76. The T-55 and PT-76 both have small caliber (100mm or less) main guns. The main gun of the T-62 is of larger caliber. The PT-76 is of low profile (height less than 2.4m) while the T-55 and T-62 have a higher profile.
>
> How can we identify any one of these tanks on the battlefield?

This problem, after some logical manipulation, can be described by the following set of *prime* rules which constitute the logical specification of the problem.

Rule 1: If the tank has a small main gun
and has low profile
then it is a PT-76.

Rule 2: If the tank has a small main gun
and has a high profile
then it is a T-55.

Rule 3: If the tank has a large main gun
and has a high profile
then it is a T-62.

It should be noted that the case of a large caliber main gun and a low profile has not been specified. This case will be considered "impossible" and will be ignored below.

In order to fully specify the logic of this problem it is also necessary to associate with each decision the possible values it may attain. For this example, these are

main gun:	**mgun**	is	small or large,
profile:	**ht**	is	high or low,
tank type:	**tank**	is	PT-76, T-55 or T-62.

If we were using a backward chaining interpreter and processed the rules in the order given above, we would obtain a procedural form of the solution which can be represented by the decision tree shown in Figure 1.

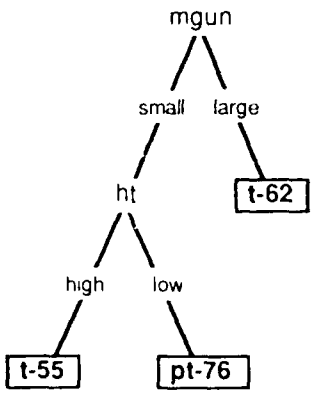


Figure 1. One procedural form for identifying Soviet tanks on the battlefield.

In all problems of practical significance, however, and even in this very simple example, multiple procedural forms of the solution are valid. An inspection of the

above rules immediately tells us that if we had ordered the antecedents differently we would have obtained the equivalent tree shown in Figure 2.

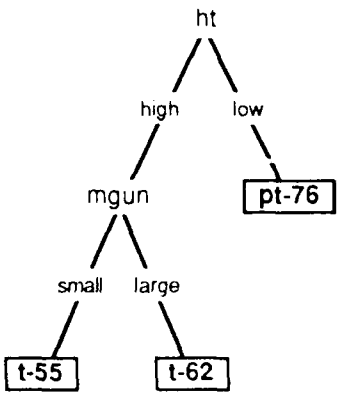


Figure 2. A rearrangement of rule antecedents yields this equivalent solution.

Which of the above solutions is more efficient? This is determined from the relative costs of making the decisions. On the battlefield, for example, it is probably easier to estimate the approximate caliber of the main gun a tank (from the size of the explosions and damage done), than to estimate its height (which requires good visibility). In this case, the solution shown in figure 1 would be preferable, since the T-62 would be identified without a need for the more "expensive" "ht" determination. The tree of figure 2 requires height estimates for all decisions.

From the above example it might be thought that obtaining a reasonable procedural form is simply a matter of sensibly ordering the rules and their antecedents. In general this is not the case. Suppose that one wanted to achieve a particular optimal behavior using prime rules and a backward chaining engine; it turns out that it may simply *not* be possible [3]. Thus, there is a nontrivial problem concerning how one turns a minimal logical specification into a procedural form.

This difficulty with imposing control in an expert system may be resolved by introducing an intermediate design step called *knowledge shaping*. The idea is to find a procedural form which simultaneously satisfies both the control and logical specifications. Knowledge shaping allows the designer to interactively explore alternative control regimes without altering the logical specification.

The idea is this: backward chaining is used as before to generate a procedural form of the knowledge. However,

rather than simply using this procedural form it is now examined to see whether it makes good procedural sense. Contrary to popular opinion, the initial procedural form, as used by an inference engine, will not necessarily make procedural sense [3]. If it does not, then the program is *shaped* into one that does using a control specification. This can be done either interactively or automatically using the algorithms based on the developments described in [1,2,5].

The advantage of this approach is that it recognizes that the abstract logical specification may not be sufficient to completely determine the procedural form. Instead, it employs a separate step which uses any control information specified to obtain a procedural version of the knowledge. This allows the logical specification to remain purely logical, and allows control information to be developed independently.

In order to illustrate these ideas, consider Table 1, a description of the main battle tanks currently deployed by the Soviet Union. In this more complex scenario, how could one best determine the type of a tank on the battlefield?

TABLE 1. PRINCIPLE SOVIET MAIN BATTLE TANKS
(extracted from Dunnigan [4])

Name	Weight	Height	Main Gun Caliber	Rate of Fire	Main Gun Range
T-80	high	low	large	high	high
T-72	high	high	large	high	avg
T-62	med	high	avg	low	avg
T-55	med	high	small	low	low
PT-76	low	low	small	low	low

One could manually encode this information in the form of rules such as:

Rule 1: If the tank has a high weight
and has a low profile
and a large main gun
and a high rate of fire
and a high main gun range
then it is a T-80.

The decisions and possible outcomes would be:

main gun range:	**mgrng**	is	high or low,
rate of fire:	**rof**	is	high or low,
main gun:	**mgun**	is	large, avg or small,
profile:	**ht**	is	high or low,
weight:	**wt**	is	high, med or low,
tank type:	**tank**	is	T-80, T-72, PT-76, T-55 or T-62.

It is important to note that if the problem logic is to be stated in rules, all attribute value combinations not specified in the table must be represented as "impossible" to ensure logical validity. In some development environments, this table can be input directly as a "training set" which greatly simplifies the logical specification of the problem. In either case, once the knowledge has been provided, it turns out that 32 efficient, decision equivalent procedural solutions exist for this relatively simple problem! Space here does not permit showing all of them. The point is that using an inference engine, the expert system would take care of these forms. The one actually used depends on the conflict resolution scheme, rule ordering, etc. With conventional expert system development tools, selection of the proper procedural form is outside of the user's direct control.

Now let's investigate what can be accomplished using knowledge shaping. The control specification is a set of rules, assignments and inequalities which deal with the relative costs of decisions, the conditional probabilities of events and the ordering of decisions. It is this type of information which makes one procedural form more attractive than another.

In our tank identification problem, some cost relationships are apparent. In a battlefield situation, it is easier to determine the firing rate and range of the tank's main gun than the gun's caliber. However, as stated previously, we can make a reasonable caliber estimate more easily than determining the tank's profile (which requires good visibility). Furthermore, determination of the weight of a tank is even more difficult. Clearly, these constraints will greatly limit the set of procedural forms which make sense. A control specification which satisfies these three constraints can be stated simply as:

mgun is worse than **rof** and **mgrng**

mgun is better than **ht**.

ht is better than **wt**.

Note that these constraints make no statement about the specific costs of the decisions. Rather, they deal only with relative terms. They state that "rof" and "mgrng" are easy to determine and that "wt" is hard to determine. This control specification reduces the list of valid procedural forms from 52 to only 8. The remaining forms are shown in Figure 3 on the following page.

The 8 solutions of figure 3 are better suited to the battlefield problem than were the other 44 which were rejected as not adhering to the decision cost guidelines. However, we can further investigate the details of our decision-making process. Although the decision "ht" appears to be significant (it appears in 7 of the 8 trees) it still necessitates good visibility, which is unlikely on a battlefield. Thus, "ht" should be avoided. Furthermore, determination of the range of a tank's main gun is a relatively easy passive activity. Thus, its use should be encouraged.

These features may be used to further constrain the solution space by making the following additions to the control specification.

use **mgrng**.

avoid **ht**

The procedural form of the solution which results is shown in Figure 4. Note that this tree is not a member of the 8 that resulted from the cost data only. The tree there had "ht" in place of "wt" (since "wt" was more expensive than "ht"). Now that the constraint to avoid "ht" has been imposed, "wt" has been promoted to replace it. The procedural form of figure 4 provides the solution to the tank identification problem which is optimal for the situation described above.

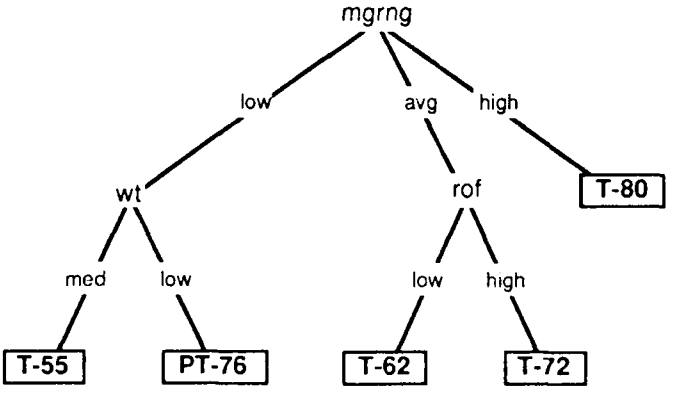


Figure 4. Imposition of a complete control specification yields the most efficient procedural solution.

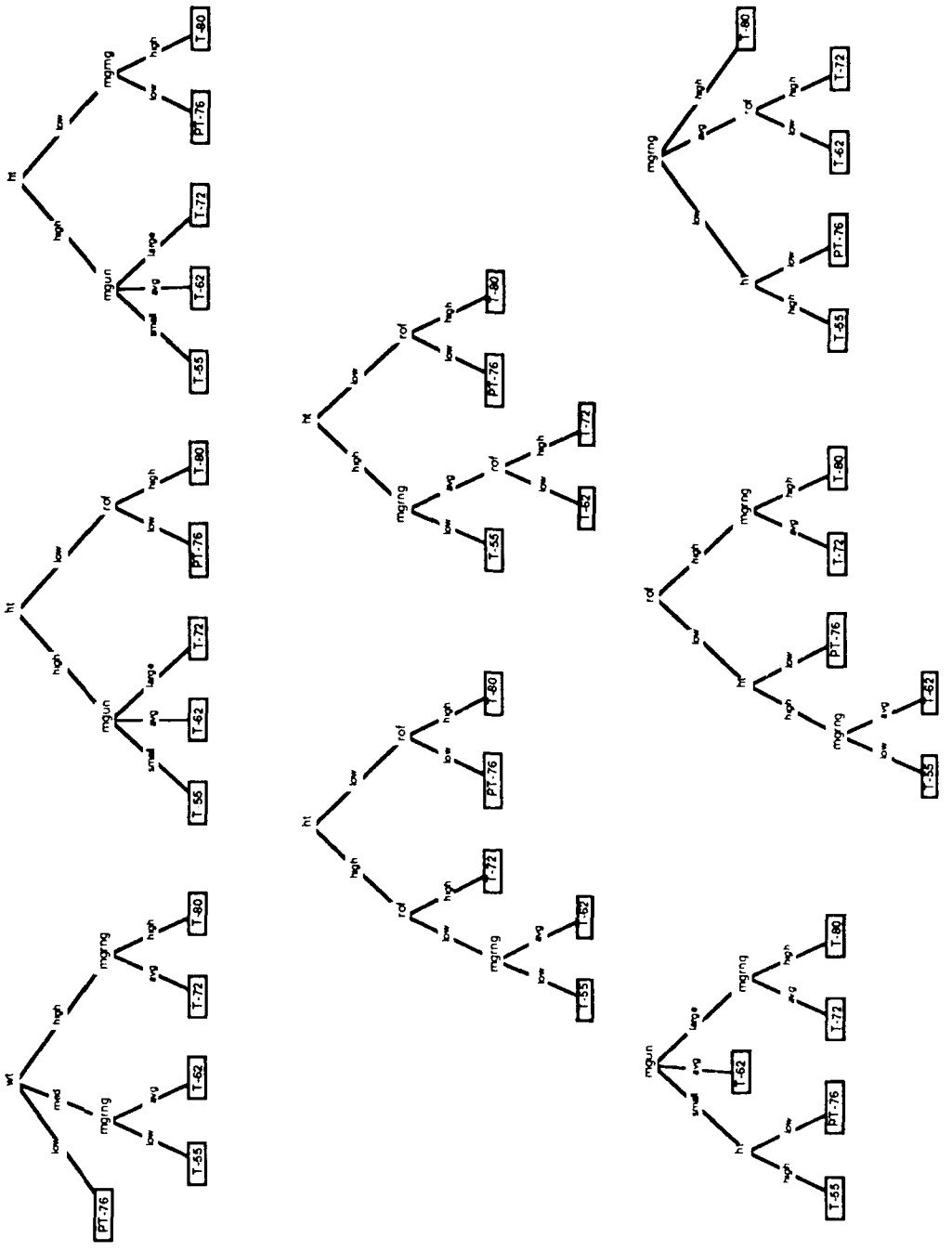


Figure 3. Using the relative costs of decisions limits the number of reasonable procedural forms for the tank identification problem.

Knowledge shaping removes the temptation for the developer to "tinker" with the system, by changing the form of the rules, in order to optimize its performance. No alteration of the logical specification is made during the knowledge shaping step; the developer can only change the control regime. The result of this step is an efficient procedural form of the expert system. This procedural form

of the knowledge is ideally suited for use in the development environment. An additional benefit of this method is that once an optimum procedural form has been generated, it may be executed directly; removing the performance problems associated with inference engines.

Is knowledge shaping feasible as a real-world approach to expert systems design and development? A version of this knowledge shaping tool has been implemented in Perceptics' Knowledge Shaper™. Using specification formats quite similar to those of the examples presented above, this software tool provides the means for approaching decision-making problems in this fashion.

REFERENCES

1. J.R.B. Cockett, "Aspects of Expert Systems," (1987). To appear in The Application of Advanced Computer Concepts and Techniques in Control Engineering, NATO Advanced Study Institute.

2. J.R.B. Cockett, "Decision Expression Optimization," *Fundamenta Informaticae*, vol. X, pp. 93-114, 1987.

3. J.R.B. Cockett and J.A. Herrera, "Prime Rule-Based Systems Give Inadequate Control". In Proceedings of the ACM SIGART ISMIS, (Knoxville, TN), pp. 441-449, 1986.

4. J.F. Dunnigan, *How to Make War*, William Morrow & Co., New York, 1983.

5. J.A. Herrera, *Theoretical Foundations and Algorithms for the Generation of Optimal Decision Trees*, Ph.D. Thesis, University of Tennessee, Knoxville, 1988.

☆U.S. GOVERNMENT PRINTING OFFICE: 1989 242-472/03830